



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE CARRERA

TÍTOL DEL TFC: Models substitutius per a modelat de dades aeronàutiques

TITULACIÓ: Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació

AUTOR: Alicia Herreros Suárez

DIRECTOR: Francisco Javier Mora Serrano

SUPERVISOR: Jordi Pons i Prats

DATA: 22 de setembre de 2015

Títol: Models substitutius per a modelat de dades aeronàutiques

Autor: Alicia Herreros Suárez

Director: Francisco Javier Mora Serrano

Supervisor: Jordi Pons i Prats

Data: 22 de setembre de 2015

Resum

Fer una simulació pot ocupar hores o fins i tot dies, per tant, és molt més eficient dissenyar i executar un nombre limitat d'aproximacions informàtiques per poder conèixer el comportament de les dades i així poder obtenir la resta dels resultats que no s'havien calculat a les simulacions de manera molt més ràpida, encara que no tant precisa.

Els mètodes d'aproximació de dades utilitzen un conjunt de punts amb uns valors coneguts per a interpolar altres punts, que estaran situats a la mateixa àrea d'estudi i dels quals no se'n coneix el valor.

La interpolació amb kriging per a predir la resposta en qualsevol punt del disseny és la idònia per a aquest estudi perquè és molt precisa, permet calcular l'error i, encara que requereix uns certs coneixements previs sobre el comportament de les dades a tractar, no precisa càlculs massa complexos.

Tenint totes aquestes idees presents, en aquest treball s'han estudiat una sèrie de dades per obtenir-ne una funció d'aproximació que faciliti el càlcul de futurs escenaris. Entenent com ajusta les dades per poder calcular valors aproximats per a qualsevol situació.

S'ha generat un model per a execucions informàtiques que realitza els càlculs amb la interpolació per kriging amb models esfèrics, exponencials i gaussians de semivariograma i mostra els resultats de manera numèrica i de manera visual amb alguns diagrames.

Després de realitzar diferents proves amb dades de diversos camps com la geologia o la medicina, s'ha comprovat que el model realitza el procés de manera molt equilibrada en quant a precisió dels resultats i temps de càlcul emprat.

Finalment s'ha aplicat a un cas amb dades reals per al predisseny de motors d'aviació, concretament per al càlcul de les toleràncies de fabricació dels àleps dels turbopropulsor.

Per últim, s'han anomenat línies de futures investigacions amb l'estudi realitzat i el programa creat com a punt de sortida.

Title: Models substitutius per a modelat de dades aeronàutiques

Author: Alicia Herreros Suárez

Director: Francisco Javier Mora Serrano

Supervisor: Jordi Pons i Prats

Date: September 22, 2015

Overview

To make a simulation can take some hours or even days, therefore, is more efficient to design and execute a limited number of computer approximations to know the behaviour of the data and be able to get the rest of the results that have not been calculated in the previous simulation. This way is much faster, although not as accurate.

The data approximation methods use a set of known values from some points to interpolate other points located in the same area of study, which value is not known.

The interpolation method by kriging to predict the response at any point of the design is suitable for this study because it's very accurate, also it allows to calculate the error, even though it requires some previous knowledge of the data to be treated, does not require very complex calculations.

Taking all these ideas present, in this work, we have studied a few data to obtain an approximation function that facilitates the calculation of future research. Understanding how to adjust the data to calculate approximate values for any situation.

We have created a model for computer executions that performs calculations with kriging interpolation for spherical, exponential and Gaussian semivariogram models, and it shows the results numerically and visually with some diagrams.

After performing different tests with different data obtained from some fields such as geology or medicine, it has been proved that the model makes the process very balanced in terms of accuracy of the results and the time of calculation used.

Finally, we have applied to a case with real data for the preliminary design of aircraft engines, specifically to calculate the manufacturing tolerances of the blades of the turboprop.

At last, we have named future research lines with this study and the program created as a starting point.

En especial a mi hermano Esteban
y su piano de cola, sin su apoyo
hubiera sido totalmente imposible.

ÍNDEX

INTRODUCCIÓ	1
CAPÍTOL 1. INTERPOLACIÓ	2
1.1. Definició.....	2
1.2. Kriging	3
1.2.1. Definició	3
1.2.2. Història.....	5
1.2.3. Tipus de kriging	5
1.2.4. Funcionament.....	7
1.2.5. Eficiència	11
1.3. Semivariograma.....	12
1.3.1. Semivariància	12
1.3.2. Comprensió del semivariograma.....	13
1.3.3. Classificació de semivariograma	14
1.3.4. Semivariograma experimental.....	15
1.3.5. Models de semivariograma	17
1.3.6. Ajust del semivariograma empíric a un model	21
1.4. Altres mètodes d'interpolació	22
1.4.1. ANN	22
1.4.2. RSM.....	23
1.4.3. IDW	23
1.4.4. Veí natural	24
1.4.5. Spline.....	24
1.4.6. Spline amb barreres	24
1.4.7. Topo to raster	25
1.4.8. Tendència	25
1.4.9. Comparació kriging amb altres mètodes.....	25

CAPÍTOL 2. MODELAT DE DADES	27
2.1. Anàlisi de dades	27
2.1.1. Dos paràmetres d'entrada	27
2.1.2. Cinc paràmetres d'entrada	28
2.2. Elecció del model i disseny del programa	30
2.3. Processos interns del programa.....	35
CAPÍTOL 3. RESULTATS OBTINGUTS.....	38
3.1. Comprovació del funcionament.....	38
3.1.1. Rang de càlcul correcte dels resultats	40
3.2. Simulacions del <i>tip clearance</i>	41
3.2.1. Simulacions amb les dades 2D	41
3.2.2. Simulacions amb les dades 5D	42
3.3. Resultats.....	42
3.3.1. Resultats 2D	42
3.3.2. Resultats 5D	44
CONCLUSIONS.....	45
BIBLIOGRAFIA	47
ANNEX 1: CODI EN JAVA	51
Main.java	51
Kriging.java	51
Coordenada.java	58
Punto.java	59
Parametros.java.....	59
Distancia.java.....	60
KrigingFrame.java	61
ANNEX 2: DADES UTILITZADES EN ELS EXEMPLES	75
Dades del nivell de cadmi.....	75
Dades de l'aquífer de Texas	76

ÍNDIX DE FIGURES I TAULES

Fig 1.1	Passos a seguir per a construcció d'aproximacions o interpolacions	3
Fig 1.2	Exemple d'interpolació per kriging amb una dimensió	4
Fig 1.3	Analogia entre 0 i 5 dimensions	9
Fig 1.4	Diagrama de les parts d'un semivariograma	13
Fig 1.5	Exemple de semivariograma experimental	15
Fig 1.6	Semivariogrames experimentals i representació matemàtica	16
Fig 1.7	Diferents models de semivariograma amb mateix abast i sostre	17
Fig 1.8	Diagrama de semivariograma esfèric.....	18
Fig 1.9	Diagrama de semivariograma exponencial	19
Fig 1.10	Diagrama de semivariograma gaussià.....	20
Fig 1.11	Diagrama de semivariograma circular.....	20
Fig 1.12	Diagrama de semivariograma lineal.....	21
Fig 2.1	Variació de la posició en la circumferència de l'anclatge del motor	27
Fig 2.2	Variació de l'angle de l'anclatge del motor	28
Fig 2.3	Gruix de la carcassa del motor	28
Fig 2.4	Posició de l'anell de rigidesa addicional	29
Fig 2.5	Gruix de l'anell de rigidesa addicional	29
Fig 2.6	Visualització del programa	31
Fig 2.7	Exemple d'un fitxer de text de dades 2D.....	32
Fig 2.8	Exemple del semivariograma empíric calculat i mostrat al programa ..	32
Fig 2.9	Exemple de semivariogrames mostrats al programa	33
Fig 2.10	Exemple de mapa de colors d'un càlcul per kriging amb s. Gaussià .	33
Fig 2.11	Exemple de result.txt.....	34
Fig 2.12	Exemple de error.txt.....	34
Fig 2.13	Exemple de la llista de pesos.....	36
Fig 2.14	Canvis en model de semivariograma sense variar dades ni fitxer	37
Fig 3.1	Interpolació amb el programa d'estudi dels nivells de cadmi	38
Fig 3.2	Comparació amb la interpolació de P. M. Barbosa	39
Fig 3.3	Interpolació amb el programa de la contaminació de l'aqüífer	39
Fig 3.4	Comparació dels resultats obtinguts amb els de N. Cressie	40
Fig 3.5	Interpolació per comprovar el rang en s. esfèric	40
Fig 3.6	Model més ajustat per a 2D	43
Fig 3.7	Model de semivariograma seleccionat	44
Taula 3.1	Resultats simulacions 2D	41
Taula 3.2	Paràmetres simulacions taula 3.1	41
Taula 3.3	Resultats simulacions 5D	42
Taula 3.4	Paràmetres simulacions taula 3.3	42

INTRODUCCIÓ

El projecte pretén aproximar un conjunt de dades i per obtenir-ne una funció d'aproximació per poder reduir el nombre de simulacions necessàries per a un model aeronàutic. Primer entenent com es poden ajustar les dades conegudes per a obtenir valors aproximats als punts desitjats i després generant un model i aplicant-lo a un cas amb dades reals.

L'objectiu principal és realitzar el càlcul d'interpolacions amb el mètode de kriging a partir d'un fitxer de dades obtingut amb una simulació o a través de mesures directes. El mètode de kriging és un mètode d'interpolació molt precís i pot proporcionar el valor de l'error, per això s'ha decidit que és el més adequat per a aquest treball.

Es volen aconseguir els resultats dels càlculs, però també és un objectiu molt important aconseguir una bona base teòrica i pràctica per a poder utilitzar en futures oportunitats, tant els coneixements obtinguts en mètodes d'interpolació com els assolits en programació amb el llenguatge orientat a objectes, Java. La base de partida per a aquesta tasca són els coneixements obtinguts a l'assignatura de Tècniques de computació i programació.

Al primer capítol s'ha fet un repàs dels conceptes de la interpolació, més concretament el mètode de kriging, definint les fórmules que s'han d'utilitzar per a realitzar els càlculs necessaris, els semivariogrames, la seva forma i la seva estructura i finalment s'ha fet una petita comparació del kriging amb altres mètodes d'interpolació diferents.

Al segon capítol s'han analitzat les dades que es volien tractar de les toleràncies de construcció dels turbopropulsors i s'ha definit el disseny del programa necessari a crear per a realitzar la interpolació per kriging, descrivint els passos que segueix per obtenir els resultats.

Al tercer capítol es mostren els resultats obtinguts de l'execució del programa per a les dades conegudes amb dos i cinc paràmetres i la precisió que s'obté en fer cadascun dels càlculs.

Finalment, s'han descrit una sèrie de conclusions assolides amb tot el procés d'elaboració del programa i el tractament dels diferents models.

També es parla de possibles ampliacions d'aquest projecte, modificant el programa creat per a adaptar-lo a altres necessitats.

És important destacar que aquest projecte el va suggerir Jordi Pons i Prats, del departament aeronàutic del CIMNE (Centre Internacional de Mètodes Numèrics en Enginyeria), el qual va oferir la possibilitat de desenvolupar-lo i així poder tenir la oportunitat de realitzar un projecte d'enginyeria amb aplicacions i dades reals.

CAPÍTOL 1. INTERPOLACIÓ

Es vol crear un model matemàtic d'interpolació de valors a partir d'uns valors que són coneguts, per això hi ha alguns conceptes que s'han de tenir clars per poder comprendre la nomenclatura utilitzada i una sèrie de fórmules que caldrà tenir en compte per poder seguir el mètode d'interpolació que es vol realitzar.

1.1. Definició

La interpolació prediu valors per als punts no mesurats a partir d'una quantitat limitada de punts de dades de mostra. Pot utilitzar-se per predir valors desconeguts de qualsevol dada en un punt geogràfic. [1]

En una interpolació se suposa que les coses més properes s'inclinen a tenir característiques similars. Per exemple, si plou a un costat del carrer, es pot preveure amb un alt nivell de confiança, que plou al altre costat del carrer, es té menys certesa sobre si plou a tot el poble, encara menys a la ciutat veïna i gairebé gens per al país on es troba el poble.

Per construir aproximacions d'anàlisi computacional s'han de seguir els següents passos: (veure **Fig 1.1**)

- 1) S'ha d'escollir un disseny experimental per provar el codi d'anàlisi informàtic.
- 2) A continuació s'ha de seleccionar un model d'aproximació per a representar les dades.
- 3) Després aquest model s'ha d'ajustar al model apropiat per a les dades de mostreig.
- 4) Per últim s'han d'aproximar les mostres amb la tècnica adient escollida.

La interpolació per kriging s'utilitza en gairebé totes les disciplines que analitzen els fenòmens de variació contínua a l'espai. Alguns dels exemples més destacats provenen de camps com l'exploració minera, la meteorologia i la epidemiologia. Encara que també s'utilitzen en aplicacions no lligades directament a la modelització de la variació de dades geoespacial, com a l'anàlisi d'imatges o de sèries temporals. És un mètode molt utilitzat en l'anàlisi espacial i els experiments computacionals.

Amb les suposicions prèvies adequades, el kriging dona la millor predicció lineal no esbiaixada dels valors intermedis sempre que es compleixin les suposicions. Els mètodes basats en altres criteris com la suavitat d'interpolació no són tan precisos perquè només calcula els valors intermedis més probables.

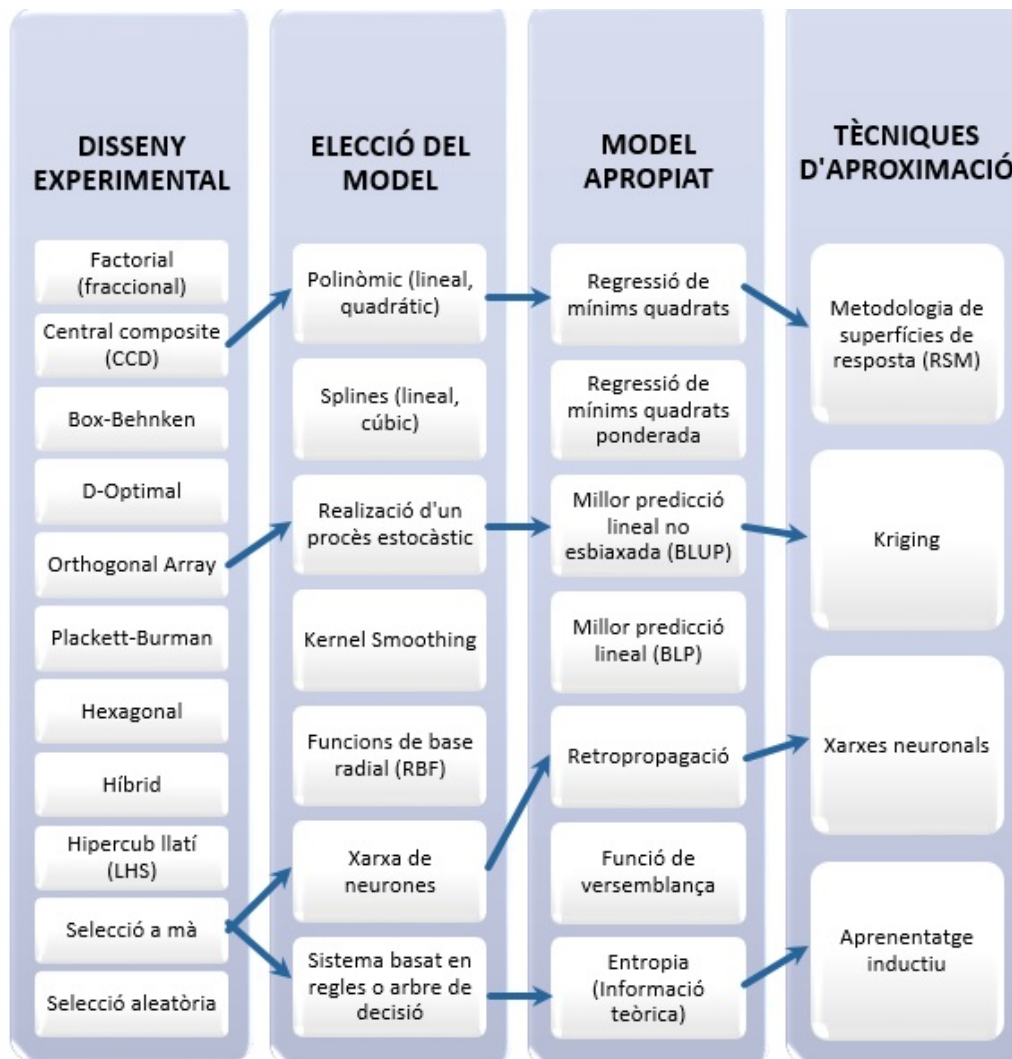


Fig 1.1 Passos a seguir per a construcció d'aproximacions o interpolacions

1.2. Kriging

1.2.1. Definició

En estadística, originalment en geoestadística, s'anomena kriging o procés de regressió Gaussiana (*Gaussian process regression*). És un mètode d'interpolació on els valors interpolats són modelats per un procés Gaussià dirigit per covariàncies seleccionades per optimitzar la suavitat dels valors ajustats.

És un procediment d'estadístiques geogràfiques avançat que genera una superfície estimada a partir d'un conjunt de punts dispersos amb valors Z . Encara més que amb altres mètodes d'interpolació, s'ha de realitzar una investigació del comportament espacial del fenomen representat pels valors Z abans de seleccionar el millor model d'estimació per a generar la superfície de sortida. [2]

Pot dir-se que és una interpolació exacta de punts, ja que els valors estimats coincideixen amb els observats, és a dir:

$$\hat{Z}(x_i) = Z(x_i) \quad (1.1)$$

Exigeix una entrada de mapa de punts i retorna un mapa amb estimacions. Pertany a la família d'algoritmes d'estimació lineal de mínims quadrats. Es considera un mètode d'estimació lineal perquè el valor obtingut és una combinació lineal, la suma ponderada, dels valors observats. (Veure apartat 1.2.4)

El kriging és un conjunt de mètodes d'interpolació espacial per estimar valors d'una variable en posicions desconegudes a partir de valors coneguts d'observacions en punts de mostreig propers. El conjunt de mètodes és estocàstic perquè tracta la variable com aleatòria.

Es basa en un algoritme generalitzat per a la millor estimació no esbiaixada, que utilitza semivariogrames per a establir les funcions de ponderació i utilitza la variància mínima dels errors d'interpolació com a criteri.

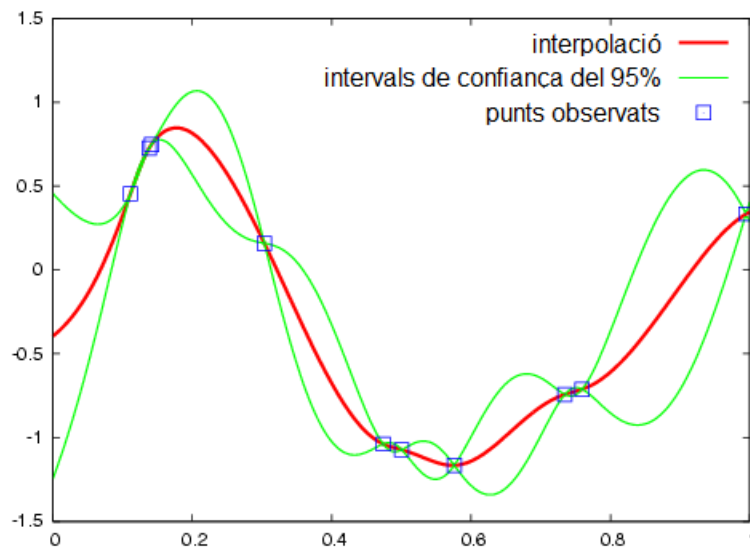


Fig 1.2 Exemple d'interpolació per kriging amb una dimensió

Té l'avantatge de ser l'únic mètode d'interpolació que proporciona l'error probable associat a les estimacions. L'error de kriging es pot observar a la equació (1.2) és el mateix que la variància de kriging i està relacionat directament amb l'anàlisi de retropropagació.

$$\gamma(\hat{Z}(x^*) - Z(x^*)) = \begin{pmatrix} w_1 \\ \dots \\ w_n \\ \mu \end{pmatrix}' \begin{pmatrix} \gamma(x_1, x^*) \\ \dots \\ \gamma(x_n, x^*) \\ 1 \end{pmatrix} \quad (1.2)$$

Per a poder utilitzar la disposició espacial en les ponderacions, la correlació espacial ha d'estar quantificada, per tant, depèn d'un model ajustat als punts mesurats, la distància a la ubicació de la predicció i les relacions espacials dels valors mesurats al voltant de la ubicació de la predicció.

1.2.2. Història

El terme kriging es va desenvolupar a partir de la tesis de màster “*A statistical approach to some mine valuations and allied problems at Witwatersrand*” al 1951 de l'enginyer de mineria Danie Gerhardus Krige, aplicada a la predicció de la localització i la qualitat dels dipòsits de mineral a Witwatersrand, a Sudàfrica. Va ser el primer en desenvolupar gràfics de la mitjana de pes-distància sobre els graus d'or basant-se en mostres d'alguns punts de la zona estudiada. [3]

El primer cop que es va utilitzar el terme kriging (en l'original francès *krigeage*) per anomenar al mètode va ser a l'estudi “*Krigeage d'un Panneau Rectangulaire par sa Périphérie*”, desenvolupat per Georges Matheron al 1960. Encara que aquest va derivar un estimador, un precursor de l'estimador de kriging, i no va utilitzar la variància. El verb en anglès es *to krige* i el nom més comú és Kriging.

L'origen i el desenvolupament del kriging i del cos teòric de la geoestadística van ser ressenyats per Cressie al 1990. [4] Encara que el concepte sembla que es va desenvolupar paral·lelament a altres disciplines com la meteorologia, per mitjà dels treballs de L. S. Gandin al 1963 a la Unió Soviètica, que va introduir un mètode d'interpolació similar anomenat Interpolació estadística o òptima. També és anomenada Predicció de Weiner-Kolmogorov gràcies a Norbert Weiner i Andrey Kolmogorov, que també van desenvolupar-lo.

Al camp de la estadística el concepte de semivariograma s'ha utilitzat també amb el nom de Funció d'estructura, a l'estudi de la probabilitat de Yaglom al 1962.

1.2.3. Tipus de kriging

1.2.3.1. Kriging simple

És el mètode més senzill de càlcul però el menys generat. S'assumeix que el valor esperat de la variable aleatòria és conegut i utilitza una funció de covariància (veure equació (1.3)) per al càlcul de la funció d'interpolació. S'ha d'assumir que hi ha estacionalitat des del primer moment a tot l'espai amb una mitjana coneguda, m com es mostra a l'equació (1.4)

$$c(x, y) = Cov(Z(x), Z(y)) \quad (1.3)$$

$$E\{Z(x)\} = E\{Z(x_0)\} = m \quad (1.4)$$

Generalment ni la covariància ni el valor esperat es són coneguts prèviament, per això és el mètode menys utilitzat. A la pràctica, el kriging simple calcula la estimació mitjançant una regressió lineal generalitzada.

1.2.3.2. Kriging ordinari

També anomenat de punts, és la variant més general i més utilitzada. El mètode ordinari ajusta variogrames de manera local a un subconjunt de punts de mostreig i determina els pesos de la funció d'interpolació a partir de la matriu de semivariàncies entre parells de punts de mostreig i del vector de semivariàncies entre els punts de mostreig i els punts que es volen estimar.

S'ha de suposar que hi ha estacionalitat intrínseca o feble de la variable aleatòria. S'ha de tenir el número suficient d'observacions per poder construir el semivariograma experimental, la mitjana de la variable aleatòria és desconeguda però és constant (equació (1.5)) i la semivariància és coneguda (equació (1.6)). Aquesta pressuposició és raonable tret que hi hagi una raó científica que ho rebutgi.

$$E(Z(x)) = \mu \quad (1.5)$$

$$\gamma(x, y) = E[(Z(x) - Z(y))^2] \quad (1.6)$$

Els pesos d'aquest mètode compleixen la restricció no esbiaixada, tal i com es pot veure en l'equació següent:

$$\sum_{i=1}^n w_i = 1 \quad (1.7)$$

1.2.3.3. Kriging universal

Assumeix un model de tendència d'invalidació de les dades. Pot modelar-se a través de la funció determinista polinòmica. Aquesta funció es resta dels punts mesurats originalment i la autocorrelació es modela a partir dels errors aleatoris. Un cop que el model s'ajusta als errors aleatoris i abans de realitzar una predicció, es torna a sumar la funció polinòmica a les prediccions per a obtenir resultats significatius.

$$E\{Z(x)\} = \sum_{k=0}^p \beta_k f_k(x) \quad (1.8)$$

Aquest tipus de kriging només s'ha d'utilitzar si es coneix una tendència a les dades i es pot donar una justificació per a descriure-la.

1.2.3.4. Altres tipus secundaris [5]

- Kriging IRFk: Assumeix que $E\{Z(x)\}$ és un polinomi desconegut en x .
- Kriging indicador: Utilitza funcions d'indicació en lloc del procés, amb la finalitat d'estimar les probabilitats de transició.
- Kriging indicador-múltiple: és una versió de l'indicador treballant amb una família d'indicadors, però no s'utilitza massa per les dificultats afegides a operar i a la validació del model.
- Kriging disjuntiu: És una generalització no lineal
- Kriging logonormal: Interpolava dades positives per mitjanes de logaritmes.

1.2.4. Funcionament

Les estimacions o prediccions es calculen com mitjanes amb un pes dels valors coneguts d'entrada. El pes dels valors s'assigna segons la distància entre els punts mesurats i la ubicació del punt que es vol saber, és a dir, tindran més pes els punts més propers que els més allunyats, i també influeix la disposició espacial general dels punts mesurats. Els punts que estan més lluny d'una distància definida tindran un pes de zero.

La estimació d'un punt de sortida, \hat{Z} , és una combinació lineal dels factors de pes, w_i , (veure apartat 1.2.4.1) i els punts d'entrada coneguts, Z_i , tal i com es pot veure en equació (1.9).

$$\hat{Z} = \sum_{i=1}^n (w_i \cdot Z_i) \quad (1.9)$$

1.2.4.1. Factors de pes

Per a calcular els valors de sortida s'han de trobar els factors de pes de cada punt d'entrada per a cada punt de sortida, després es multiplicarà cada pes pel corresponent punt d'entrada i es sumaran tots els resultats calculats.

Els pesos intenten resumir dos processos molt importants en el procés d'inferència espacial:

- 1) Reflexar la proximitat estructural de les mostres a la localització de la estimació, x_0 .
- 2) A l'hora, tenir un efecte de no segregació per evitar el biaix causat per conglomerats de la mostra eventuals.

El càlcul del factor de pes té dos objectius clars que són: no esbiaixar i que la variància d'estimació sigui mínima.

Els factors de pes no només són determinats per les distàncies als punts d'entrada amb respecte als de sortida, sinó que es calculen trobant els valors del semivariograma per a cada distància entre el punt de sortida i cada punt d'entrada, aleshores una sèrie d'equacions simultànies s'han de resoldre, per aconseguir que, com s'ha dit abans, l'error de sortida sigui mínim.

1.2.4.2. Calcular la distància entre parells de punts

El mètode original va ser formulat per problemes d'una, dues o tres dimensions que es trobaven a l'àrea de la geoestadística minera. Però amb el propòsit d'optimitzar la funció s'utilitza una versió general L-dimensional (veure **Fig 1.3**). [6] La distància per L dimensions entre dos punts es realitza amb l'equació (1.11). Cal dir que amb el vector X es representa un punt concret a l'espai, és a dir, un dels punts d'entrada (o de sortida), com es pot veure a l'equació (1.10).

$$X = \{x_1, x_2, x_3, \dots, x_L\} \quad (1.10)$$

$$h = \sqrt{(x_{11} - x_{12})^2 + (x_{21} - x_{22})^2 + (x_{31} - x_{32})^2 + \dots + (x_{L1} - x_{L2})^2} \quad (1.11)$$

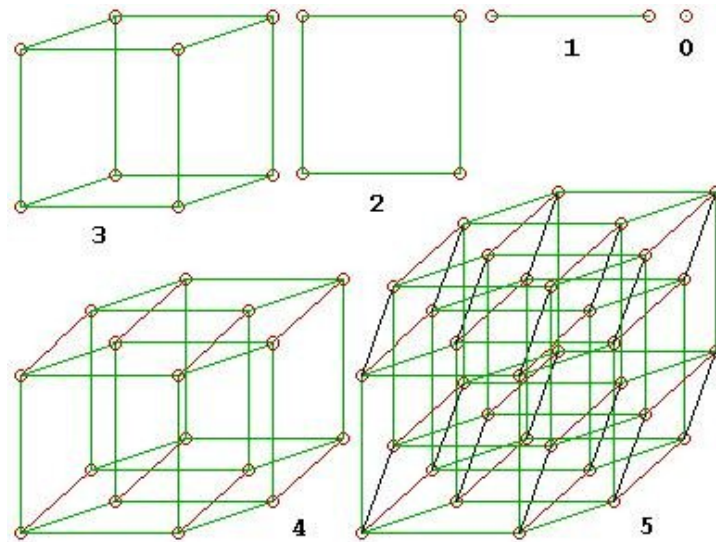


Fig 1.3 Analogia entre 0 i 5 dimensions

1.2.4.3. Calcular una interpolació amb el mètode de kriging

Amb una sèrie de mostres n , en un espai concret, cada mostra és $Z(x_n)$, la sèrie de mostres aleatòries es $Z(x_i)$ i la correlació entre dues mostres depèn de la distància entre elles, és independent de la seva localització.

$$\hat{Z}(x_0) = [w_1 \quad w_2 \quad \dots \quad w_n] \cdot \begin{bmatrix} Z_1 \\ Z_2 \\ \dots \\ Z_n \end{bmatrix} = \sum_{i=1}^n w_i(x_0) \times Z(x_i) \quad (1.12)$$

Per realitzar una predicció amb aquest mètode d'interpolació és necessari descobrir les regles de dependència. S'han de crear els variogrames i les funcions de covariància per a calcular els valors de dependència estadística, anomenada també autocorrelació espacial. Depenen del model d'autocorrelació, és a dir, s'ha d'ajustar un model matemàtic, i després es calcularan els valors desconeguts, o el que és el mateix, es farà la predicció.

S'han de seguir una sèrie de passos concrets per obtenir el resultat desitjat de la predicció: [7]

- 1) S'han de trobar els punts vàlids d'entrada, és a dir, s'han de descartar els punts que no tenen unes coordenades definides, o que no tenen un valor o que estan duplicats. Si estan duplicats es pot realitzar la mitjana, es pot negligir o fer servir el primer d'aquests, però sempre amb el mateix criteri per a tots els casos de duplicat.
- 2) Es determinen les distàncies, h , entre tots els punts d'entrada vàlids i s'ordenen de menor a major.

- 3) S'ha de calcular la semivariància $\hat{\gamma}(h)$ per a cadascuna de les distàncies obtingudes (veure apartat 1.3.1).
- 4) Es col·loquen els valors del semivariograma en un diagrama amb la correlació espacial per a poder determinar quin és el model que més s'ajusta al semivariograma empíric, saber quin és el residu, quin és l'abast i el sostre (veure apartat 1.3.2).
- 5) Un cop s'elegeix el model de semivariograma, s'ha de calcular el valor del semivariograma per a cada parell de punts i omplir la matriu V mostrada en l'equació **(1.13)**, que serà de $(n+1) \times (n+1)$.

$$V = \begin{pmatrix} 0 & \gamma(h_{12}) & \gamma(h_{13}) & \dots & \gamma(h_{1n}) & 1 \\ \gamma(h_{21}) & 0 & \gamma(h_{23}) & \dots & \gamma(h_{2n}) & 1 \\ \gamma(h_{31}) & \gamma(h_{32}) & 0 & \dots & \gamma(h_{3n}) & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \gamma(h_{n1}) & \gamma(h_{n2}) & \gamma(h_{n3}) & \dots & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad (1.13)$$

- 6) Després es calcula la inversa de la matriu V i es reserva per a un càlcul posterior.
- 7) Per al punt de sortida seleccionat, es calculen també les distàncies amb cadascun dels punts d'entrada, i es troba el valor del semivariograma seleccionat abans, s'omple el vector D (equació **(1.14)**), de tamany $n+1$.

$$D = \begin{pmatrix} \gamma(h_{p1}) \\ \gamma(h_{p2}) \\ \gamma(h_{p3}) \\ \dots \\ \gamma(h_{pn}) \\ 1 \end{pmatrix} \quad (1.14)$$

- 8) Es calculen els factors de pes per al punt de sortida que s'està calculant, multiplicant la matriu inversa de V calculada al pas 6 per D , trobat al pas 7 (equació **(1.15)**). Els factors de pes calculats només seran vàlids per un al de sortida concret.

$$w = V^{-1} \cdot D \quad (1.15)$$

- 9) Per calcular el valor de la predicció per al punt de sortida, s'han de sumar els productes dels factors de pes pels valors d'entrada, tal i com es veia a l'equació **(1.9)**.

- 10) Es repeteixen els passos des del 7 fins al 9 per a cada punt de sortida que es vulgui saber, \hat{Z} .

En lloc d'agafar tots els punts per a calcular el semivariograma i omplir la matriu V , també es poden escollir només els punts més propers al punt de sortida, ja que són els que tindran més influència sobre aquest, els punts més allunyats s'ignoraran. Es determinarà un nombre de punts mínim necessaris i s'agafarà aquest nombre de punts d'entrada començant des del més proper al de sortida que es vol calcular i a partir del pas 5 es realitzarà tot amb la quantitat de punts desitjada.

Es podria dir que amb el mètode de kriging, les dades s'utilitzen dues vegades, degut a les dues tasques ben distintives que es poden trobar: el primer cop es faran servir per a calcular l'autocorrelació espacial de les dades i la segona vegada per a calcular les prediccions.

Per a poder implementar el mètode de kriging, els punts que són més lluny d'una certa distància definida tindran un pes de zero, ja que sinó, el càlcul seria impossible, degut al temps que es trigaria, és a dir, els punts que són més llunyans no afectaran al resultat.

1.2.5. Eficiència

Fer una simulació amb el mètode d'element finit pot trigar hores o fins i tot dies, per tant, és molt més eficient dissenyar i executar un nombre limitat de simulacions informàtiques i aleshores utilitzar la interpolació amb kriging per a predir la resposta en qualsevol dels altres punts de disseny.

Com que es necessiten mostres per a generar la superfície, el cost de creació creix exponencialment amb el nombre de mostres i s'ha de trobar l'equilibri entre precisió i cost.

Per a crear una superfície contínua del fenomen estudiat es realitzen prediccions per a cada ubicació en l'àrea d'estudi basades en el semivariograma i la disposició espacial dels valors mesurats que són propers.

Es pot demostrar que l'ús de models de kriging com alternatives a altres mètodes de superfície de resposta per a construir una aproximació global precisa és bo, ja que es facilita la optimització i és una mica més precís que els altres mètodes. [8]

Quan es representa el núvol de valors reals en front als estimats, la mitjana de les estimacions ha de ser igual o molt similar a la dels valors reals i la desviació al quadrat ha de ser mínima perquè l'estimació sigui precisa.

1.3. Semivariograma

Les prediccions de kriging poden ser expressades en termes de variograma en lloc de covariància. Tots els semivariogrames es calculen utilitzant el model definit per l'usuari.

Un semivariograma és un mètode exploratori en forma de gràfic utilitzat sobretot en geoestadística per a representar una funció de semivariància que descriu el grau de dependència o correlació espacial d'una variable aleatòria o procés estocàstic, distribuïda de forma contínua a l'espai. Proporciona una descripció quantificada de la variació regionalitzada d'una variable espacial aleatòria i permet identificar els patrons de variabilitat espacial, optimitzar la separació i ajustar un model de semivariograma que serveix per a la interpolació espacial de la variable analitzada. [9]

La modelització per mitjà de semivariogrames pot tenir errors i no és necessàriament la millor modelització possible per un determinat conjunt d'observacions d'una variable. El principal avantatge del semivariograma és la capacitat d'exploració de la variació espacial que proporciona, per obtenir una visió del conjunt sobre les propietats de la variació espacial corresponent a unes dades de mostra.

Per comoditat s'anomena semivariograma tant a la funció de semivariància de les diferències de valors entre posicions segons la distància, com al gràfic que la representa. S'utilitza molt sovint incorrectament el terme variograma com un sinònim de semivariograma. [10]

1.3.1. Semivariància

L'ajust d'un model o modelat espacial, també és conegut com anàlisi estructural o variografia. En el modelat espacial de l'estructura dels punts mesurats, es comença amb un gràfic del semivariograma empíric, calculat per a tots els parells d'ubicacions separats per la distància h . Sovint cada parell d'ubicacions té una distància única i normalment hi ha diversos parells de punts. La diagramació de tots els parells ràpidament es torna impossible d'administrar i els parells es poden agrupar per intervals.

La semivariància de les diferències de valor entre posicions és una de les funcions més significatives per poder analitzar la correlació espacial entre els valors d'una variable mesurada amb un conjunt de punts de mostreig.

La funció es defineix com la meitat de la variància de les diferències dels valors de la variable aleatòria entre parells de posicions separades per una determinada distància h . En el cas de que algunes distàncies estiguin repetides n vegades es realitzarà la mitjana de les diferències dels valors al quadrat, equació **(1.16)**.

$$\hat{\gamma}(h) = \frac{1}{2 \cdot n(h)} \sum_{i=1}^{n(h)} (z(x_i + h) - z(x_i))^2 \quad (1.16)$$

1.3.2. Comprensió del semivariograma

Un cop tots els parells d'ubicacions han estat col·locats en el diagrama es pot ajustar un model (veure apartats 1.3.5 i 1.3.6). Per descriure els models s'utilitzen característiques: el sostre, l'abast i el residu. [10]

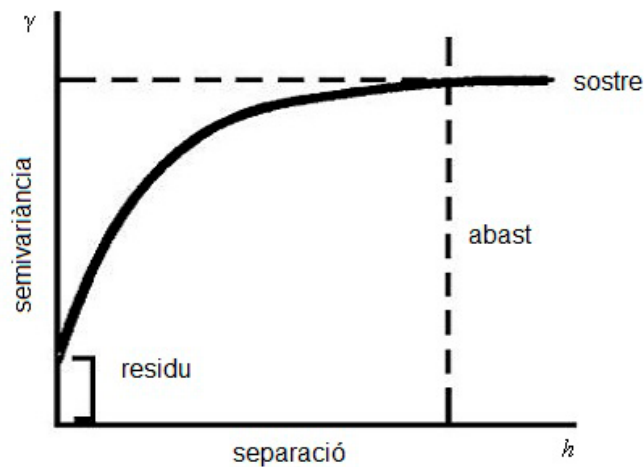


Fig 1.4 Diagrama de les parts d'un semivariograma

1.3.2.1. Abast

També anomenat *range*, definit per la lletra *a* en les equacions. En observar el model d'un semivariograma, es pot veure que a una determinada distància, el model s'anivella. La distància on comença a ser més pla el model es denomina abast. Això és perquè les mostres més juntes tenen influència unes sobre les altres, mentre que les que estan més allunyades no exerceixen gairebé influència.

1.3.2.2. Sostre

També s'anomena *sill* i es representa amb *c* a les equacions. És el valor on s'assoleix l'abast, l'eix Y (valor de la semivariància). Un sostre parcial serà el sostre menys el residu.

1.3.2.3. Residu

També anomenat *nugget* i expressat com a c_0 a les equacions. Quan el semivariograma s'extrapola fins a la distància zero, idealment el valor de la semivariància hauria de ser zero també. Així mateix, factors com l'error de mesura o de mostreig, poden ocasionar que el valor de la semivariància a distància infinitament inferior sigui un valor no nul, anomenat residu. Si el model de semivariograma intercepta l'eix Y en un punt r , aquest punt r serà el valor del residu.

Els errors de mesura succeeixen degut a l'error dels dispositius de mesura. Els fenòmens naturals poden variar espacialment en un rang d'escala, la variació a micro-escala més petites que les distàncies de mostreig, apareixeran com a part de l'efecte residu.

Abans de recopilar dades és important comprendre les escales de variació espacial en què s'està interessat.

1.3.3. Classificació de semivariograma

Els semivariogrames tenen diferents classificacions possibles segons els criteris que es segueixin. [11]

Es poden classificar segons les propietats d'isotropia:

- Isòtrop: mostrarà una variació espacial igual en totes les direccions, és el més comú.
- Anisòtrop: tindrà un model de variació espacial que serà diferent segons la direcció, és a dir, serà direccional.

També es poden classificar segons la transivitativat;

- Transitiu: posseirà un abast i un sostre que es trobaran dins del domini d'interès estudiat, és el més comú.
- Intransitiu: el sostre no serà dins del domini d'interès, sinó que estarà més allunyat, per tant l'abast tampoc hi serà a la zona d'estudi.

Per últim pot ser diferent la seva escala:

- Multi-escala: es podrà descomposar en diversos components de variació espacial, segons un model lineal de variables regionalitzades.

1.3.4. Semivariograma experimental

El semivariograma elaborat a partir de les observacions, o semivariograma experimental o empíric, és un gràfic de la mitjana dels valors de semivariograma a l'eix Y, i la distància o interval a l'eix X.

A mesura que els parells d'ubicacions estiguin més separats entre ells, cap a la dreta de l'eix X, haurien de ser més diferents entre ells i tenir una diferència quadrada més gran, és a dir, cap amunt a l'eix Y, com es pot observar a la **Fig 1.5**.

És molt útil per construir models matemàtics o probabilístics que descriguin la variació espacial de la variable d'interès, entre punts de mostreig en forma de mesura de variació dels valors en funció de la distància. A més de descriure la variació espacial de la variable, serveixen per fer estimacions o interpolacions espacials de valors de la variable en posicions desconegudes, principalment en el mètode d'interpolació de kriging que és el model que s'estudia aquí.

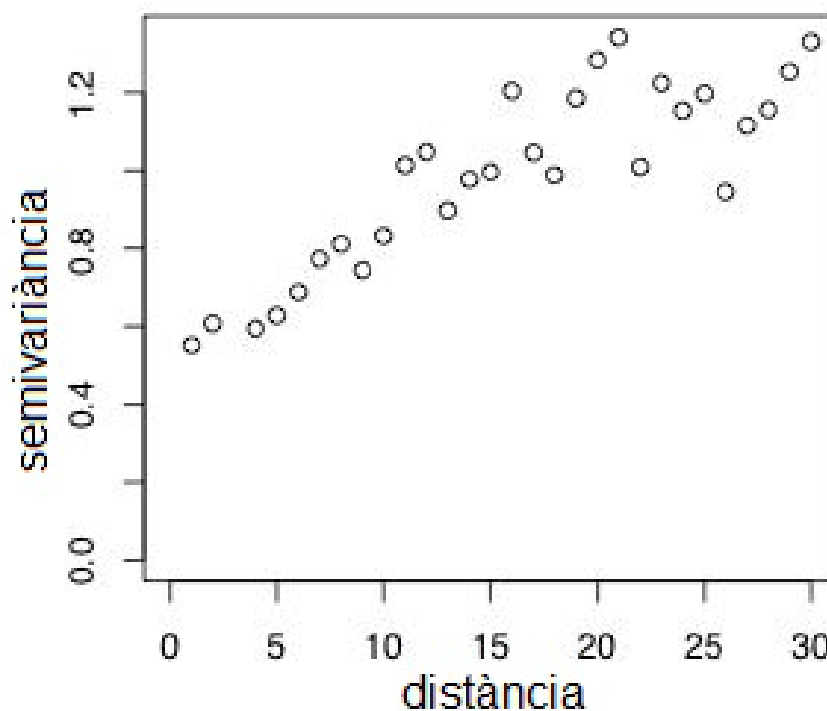


Fig 1.5 Exemple de semivariograma experimental

És necessari calcular el semivariograma empíric i després aplicar-hi el model teòric que més s'apropa. Observant la forma del semivariograma es pot veure com la funció de semivariància de les diferències entre els valors de la variable augmenta en funció de la separació fins a arribar a una tendència asimptòtica, al valor del sostre i de l'abast.

Aquest punt permet descriure i mesurar la variabilitat i ajustar un model de funció per a realitzar la interpolació o estimació dels valors en posicions on el resultat és desconegut. A la **Fig 1.6** es poden veure diferents exemples de semivariogrames empírics on s'ha aplicat el model teòric més proper.

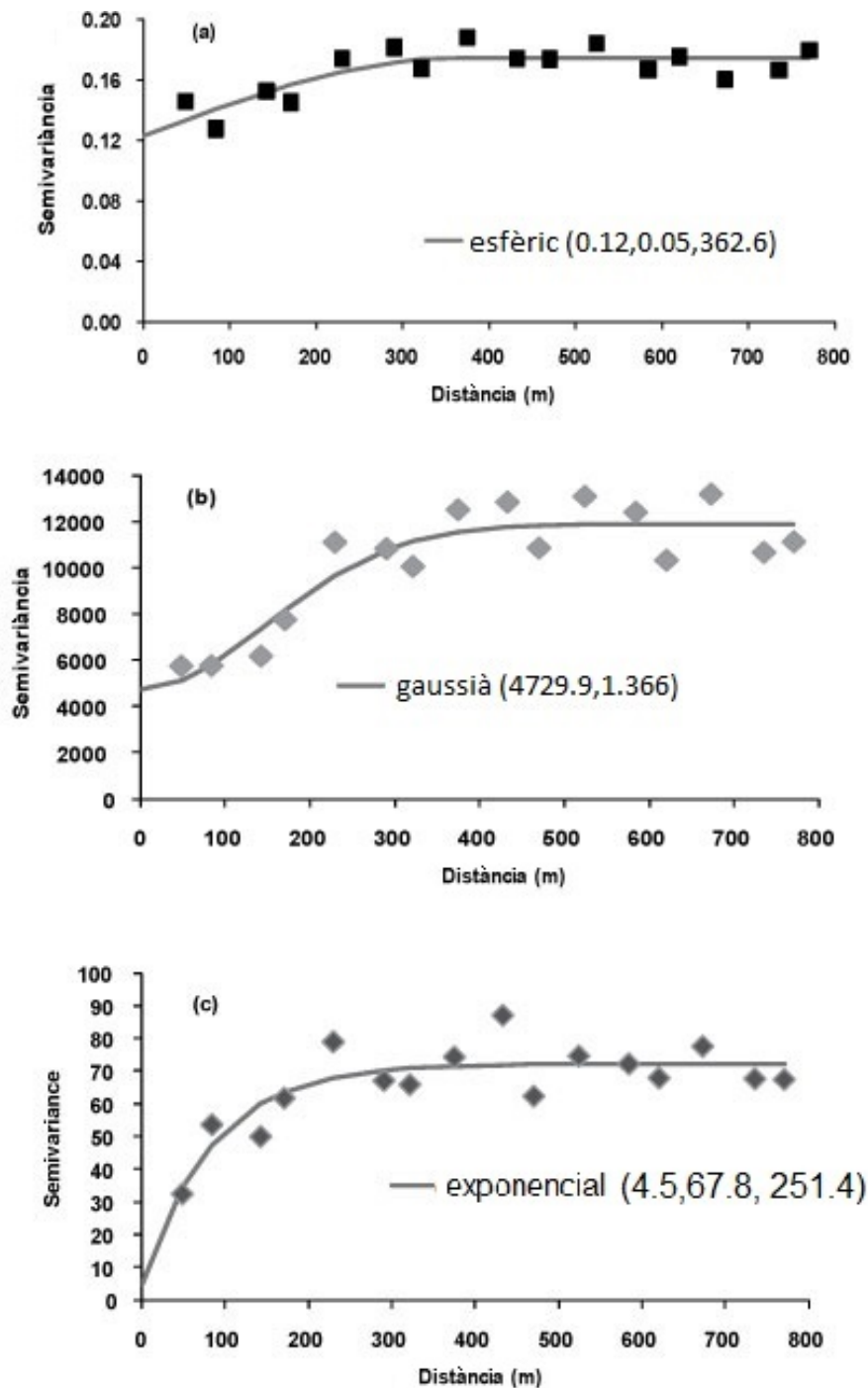


Fig 1.6 Semivariogrames experimentals i representació matemàtica

1.3.5. Models de semivariograma

Un cop es disposa del semivariograma experimental constituït a partir de les observacions dels punts de mostreig, l'anàlisi descriptiu de la variació espacial de la variable resultat o la interpolació de valors en posicions desconegudes, necessita l'ajust d'una funció matemàtica als valors de semivariància i separació observats en el gràfic del semivariograma. La funció matemàtica ajustada s'anomena model de semivariograma i utilitza diversos tipus de funcions, entre els quals el model esfèric correspon a l'ideal de variable regionalitzada i el model gaussià correspon a la distribució de probabilitat normal. En alguns casos amb variacions complexes s'utilitzen combinacions de diversos models.(veure [12])

Els models bàsics de semivariograma només consideren la separació entre els parells de punts de mostreig a l'hora de modelitzar la variació dels valors, però no la direcció relativa entre cada parell de punts. Es a dir, assumeixen que la variació compleix la condició d'isotropia.

Lògicament, havent-hi diversos models possibles, es tracta d'escollir el que resulti més adequat a les característiques observades, ja que el model seleccionat influeix en la predicció dels valors desconeguts, en particular quan la forma de la corba propera a l'origen difereix significativament. Si la corba propera a l'origen és més pronunciada, més influiran els punts propers en la predicció. Com a resultat, la superfície de sortida serà menys suau.

Cada model està dissenyat per ajustar-se a diferents tipus de fenòmens de forma més precisa.

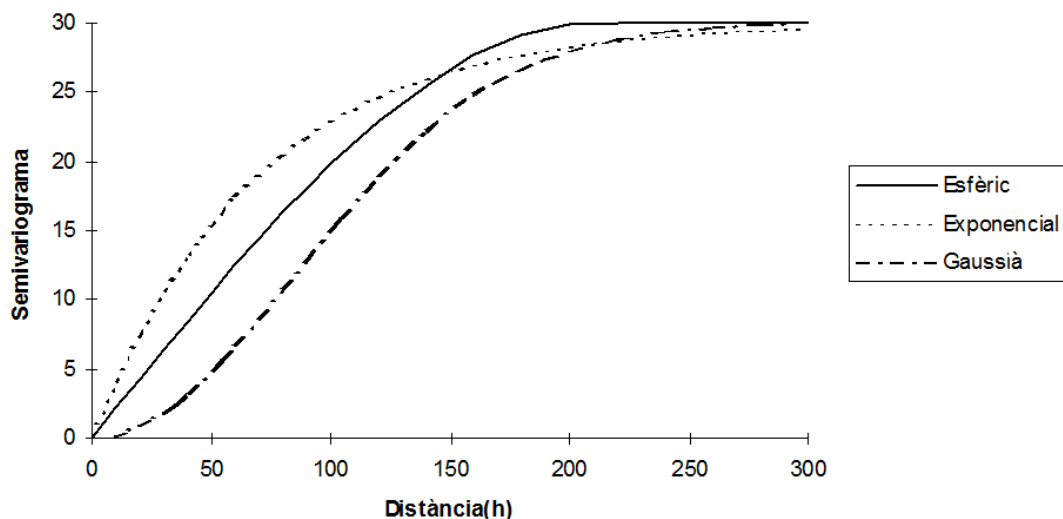


Fig 1.7 Diferents models de semivariograma amb mateix abast i sostre

Un expert en el domini del fenomen estudiat pot arribar a seleccionar el model més apropiat comparant visualment la forma de la funció que descriuen els punts del semivariograma experimental amb la forma i propietats de cadascun dels diferents models teòrics (veure apartat 1.3.5). Encara que aquest mètode és sovint massa aventurat i requereix un coneixement a fons del domini, l'examen del semivariograma experimental continua essent un procediment exploratori necessari i el punt de partida obligatori abans d'emprendre la selecció i ajust del model per mitjà de mètodes que ofereixen una mesura de control ajustada.

1.3.5.1. Semivariograma esfèric

El model esfèric de semivariograma es basa en una funció esfèrica per ajustar una funció al semivariograma experimental que resulta de les dades observades.

També és anomenat model de Matheron, i es considera que té la forma ideal perquè l'abast i el sostre són clarament visibles, també el residu es pot veure de manera clara.

Es mostra una disminució progressiva de l'autocorrelació i un augment de la semivariància fins a una certa distància, la distància de l'abast, després de la qual l'autocorrelació és zero. Que serà quan la corba es manté constant.

$$\begin{aligned} \gamma(h) &= c_0 + c \left(\frac{3h}{2a} - \frac{1}{2} \left(\frac{h}{a} \right)^3 \right) & 0 < h < a \\ \gamma(h) &= c_0 + c & h > a \\ \gamma(0) &= 0 \end{aligned} \quad (1.17)$$

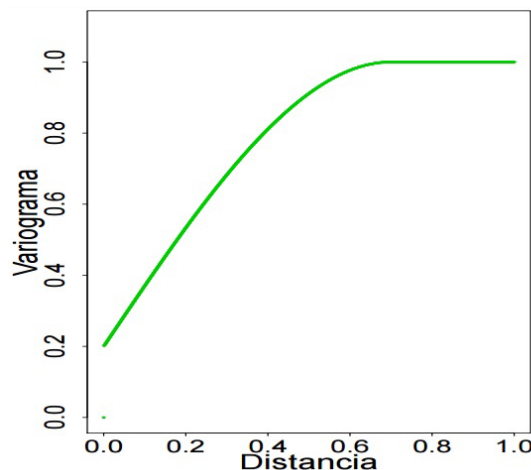


Fig 1.8 Diagrama de semivariograma esfèric

1.3.5.2. Semivariograma exponencial

El model exponencial de semivariograma ajusta una funció exponencial al semivariograma experimental. Aquest model té un residu visible i un sostre al qual s'arriba assintòticament i l'abast és gradual, és a dir, continua pujant indefinidament.

S'aplica quan l'autocorrelació espacial disminueix exponencialment en augmentar la distància. L'autocorrelació desapareix completament només a una distància infinita.

$$\begin{aligned}\gamma(h) &= c_0 + c \left(1 - \exp\left(\frac{-h}{a}\right) \right) & h > 0 \\ \gamma(0) &= 0\end{aligned}\tag{1.18}$$

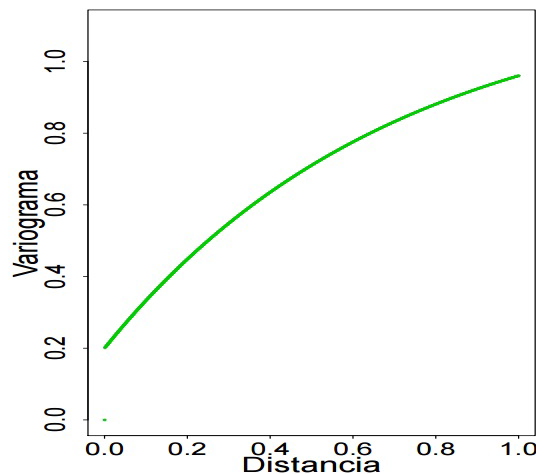


Fig 1.9 Diagrama de semivariograma exponencial

1.3.5.3. Semivariograma gaussià

El model de semivariograma gaussià ajusta una distribució de probabilitat normal al model experimental. Tindrà un residu visible, el sostre serà relativament visible i l'abast serà progressiu.

S'aplica quan l'autocorrelació espacial molt propera no és massa gran i després augmenta ràpidament fins a una certa distància que serà l'abast on s'estabilitzarà de nou.

$$\begin{aligned}\gamma(h) &= c_0 + c \left(1 - \exp\left(-\frac{h^2}{a^2}\right) \right) & h > 0 \\ \gamma(0) &= 0\end{aligned}\tag{1.19}$$

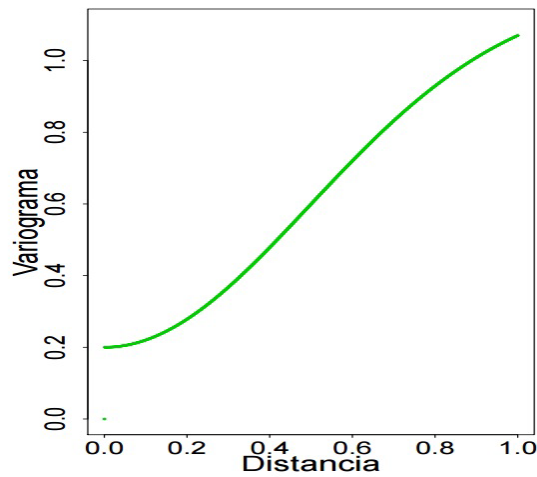


Fig 1.10 Diagrama de semivariograma gaussià

1.3.5.4. Semivariograma circular

El model circular té un residu visible però l'abast i el sostre tenen una tendència circular, és a dir, pugen de manera conjunta fins a un punt i després baixa el sostre mentre que l'abast continua pujant. És el menys utilitzat de tots.

$$\begin{aligned} \gamma(h) &= c_0 + c \left(1 - \frac{2}{\pi} \cos^{-1} \left(\frac{h}{a} \right) + \sqrt{1 - \frac{h^2}{a^2}} \right) & 0 < h < a \\ \gamma(h) &= c_0 + c & h > a \\ \gamma(0) &= 0 \end{aligned} \quad (1.20)$$

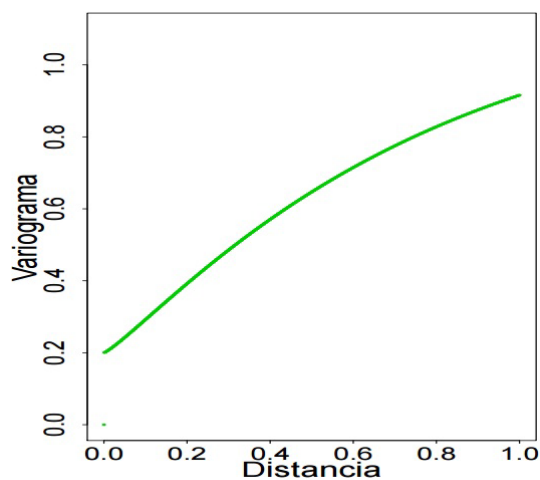


Fig 1.11 Diagrama de semivariograma circular

1.3.5.5. Semivariograma lineal

Per a distàncies petites, hi ha un comportament lineal. La correlació espacial dependrà de la recta a l'origen, si hi ha més pendent hi haurà més correlació, si la pendent no és tan inclinada, n'hi haurà menys.

En una distància determinada hi haurà zero correlació espacial, que serà al valor de l'abast.

$$\begin{aligned} \gamma(h) &= c_0 + c \left(\frac{h}{a} \right) & 0 < h < a \\ \gamma(h) &= c_0 + c & h > a \\ \gamma(0) &= 0 \end{aligned} \quad (1.21)$$

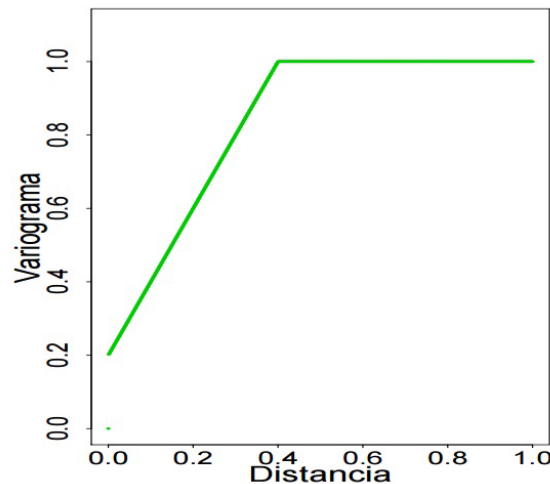


Fig 1.12 Diagrama de semivariograma lineal

1.3.6. Ajust del semivariograma empíric a un model

Ajustar un models als punts que forma el semivariograma empíric és un pas clau entre la descripció espacial i la predicció espacial. El semivariograma empíric proporciona informació sobre l'autocorrelació espacial de les dades, no obstant, no subministra informació per a totes les direccions i distàncies possibles, per això és necessari ajustar-lo a un model.

És essencialment un problema d'ajustar la millor funció possible al conjunt de valors de semivariància i separació calculats empíricament, és a dir, un problema d'ajust de corbes a un conjunt de punts.

Els mètodes que s'utilitzen per ajustar el model als punts del semivariograma poden ser diferents, i estan dividits en dos grans grups. [13]

- Els mètodes no paramètrics, són aquells que no estan ajustats als criteris paramètrics. Són determinats per les dades observades.
 - Ajust de mínims quadrats: segueix la lògica clàssica d'ajust de corbes, efectua l'ajust de mínims quadrats per cadascun dels models teòrics del semivariograma i comprova la qualitat de l'ajust de cada model per acabar seleccionant el que en presenta una millor.
- Els mètodes paramètrics són determinats per un número concret de paràmetres. Hi ha alguns tipus:
 - Màxima probabilitat: aquest és un mètode d'ajust que assumeix que la distribució del model de la funció que segueixen les dades és coneguda i calcula els valors dels paràmetres que defineixen aquesta distribució de probabilitat per les dades, de manera que els valors dels paràmetres del model seran els més probables.
 - Màxima probabilitat restringida: ja que el de màxima probabilitat és un mètode de càlcul molt intensiu, va proposar-se aquest, que redueix les necessitats de càlcul però és similar.
 - Validació creuada: mètode que permet avaluar el model de semivariograma seleccionat. Abans d'ajustar el model s'eliminen alguns punts i s'ajusta el model sense utilitzar aquests punts, i després s'aplica el model ajustat per estimar els valors en els punts eliminats. Si la diferència entre els valors estimats i els originals d'aquests punts és baixa, el model estarà més ajustat.

Un cop s'ha descobert la dependència entre les dades i s'ha ajustat el model, el semivariograma empíric s'aparta, ja que la seva funció ja ha estat completa del tot, només era necessari per a poder determinar el model matemàtic més adient per al cas estudiat.

1.4. Altres mètodes d'interpolació

1.4.1. ANN

La xarxa neuronal artificial o ANN simula el comportament del cervell. Els models exploren moltes hipòtesis simultàniament, utilitzant xarxes paral·leles formades per elements de processament connectats a través d'enllaços amb pesos. La xarxa es defineix amb valors d'entrada, els pesos associats a les connexions, el potencial d'activació, una funció d'activació i l'equació de la xarxa d'entrada.

Es pren un valor inicial per als paràmetres correctes, s'observa la sortida que s'obté i es compara amb la sortida desitjada, que és l'error. Amb un algoritme adequat es modifiquen els paràmetres en funció de l'error.

Utilitzen algoritmes d'aprenentatge adaptatiu i auto-organització que ofereixen millors possibilitats de procés robust i adaptatiu, augmenten la capacitat de la xarxa per a aproximar funcions, classificar patrons.

L'element bàsic de computació, anomenat node, rep un *input* des d'altres nodes o des d'una font externa de dades. Cada *input* té un pes associat que va modificant l'anomenat procés d'aprenentatge. Cada node aplica una funció donada de la suma dels *inputs* ponderats a través dels pesos. El resultat pot servir com a entrada d'altres unitats.

A la fase d'entrenament s'utilitza un conjunt de dades o patrons d'entrenament per a determinar els pesos o paràmetres que defineixen el model de xarxa neuronal. Es calculen de manera iterativa d'acord amb els valors d'entrenament. Com que pot ser que el model s'ajusti massa a les particularitats presents en els patrons d'entrenament, perdent la seva habilitat de generar el seu aprenentatge a nous casos, és aconsellable utilitzar un segon grup de dades diferents als d'entrenament que permeti controlar el procés d'aprenentatge.

Per obtenir els pesos s'optimitza alguna funció d'energia. Un criteri molt utilitzat és minimitzar l'error quadràtic mitjà entre el valor de sortida i el valor real esperat. [14]

1.4.2. RSM

La metodologia de la superfície de resposta o RSM és un conjunt de tècniques matemàtiques i estadístiques utilitzades per a modelar i analitzar problemes en els que una variable d'interès és influenciada per altres.

El propòsit d'aquestes tècniques és dissenyar un experiment que proporcioni valors raonables de la variable resposta gràcies a la relació que té amb les variables de predicció i determinar el model matemàtic que millor s'ajusta a les dades obtingudes. També es volen establir els valors dels factors que optimitzen el valor de la variable resposta. Això s'aconsegueix en determinar les condicions òptimes d'operació del sistema. [15]

S'utilitza perquè és fàcil d'aplicar i d'estimar encara que es sàpiga ben poc del procés.

1.4.3. IDW

La ponderació de distància inversa o IDW, utilitza un mètode d'interpolació que estima els valors calculant mitjanes dels valors dels punts de les dades de mostra al voltant de cada punt de processament. Si un punt és més a prop del que s'està estimant, més influència o pes tindrà en el procés de càlcul de la mitja. La ponderació depèn exclusivament de la distància a la ubicació de la predicció.

Els millors resultats s'obtenen si la mostra és suficientment densa respecte de la variació local que intenta simular. Si la mostra dels punts d'entrada és dispersa, els resultats poden no representar en forma suficientment adient la superfície desitjada. [2]

1.4.4. Veí natural

El mètode de veí natural troba el subconjunt de mostres d'entrada més proper a un punt de consulta i aplica ponderacions sobre aquestes basant-se en àrees proporcionals per a interpolar un valor (Sibson, 1981), també es coneix com interpolació de Sibson o de "robatori d'àrea". És local, utilitza només un subconjunt de mostres circumdants a un punt de consulta i assegura que les altures interpolades estaran dins del rang de les mostres utilitzades. És una superfície suau en totes les parts, excepte en les ubicacions de les mostres d'entrada.

Els veïns naturals d'un punt són aquells associats amb els polígons de Voronoi adjacents. Es construeix un diagrama de Voronoi de tots els punts donats, representats per polígons, després es crea un nou polígon al voltant del punt d'interpolació. La proporció de superposició s'utilitza com a ponderacions. [2]

1.4.5. Spline

El mètode spline utilitza un mètode d'interpolació que estima valors utilitzant una funció matemàtica que minimitza la curvatura general de la superfície, el que resulta en una superfície suau que passa exactament pels punts d'entrada. És una corba diferenciable definida en porcions mitjançant polinomis

Ajusta una funció matemàtica a una quantitat especificada de punts d'entrada més propers mentre que passa a través dels punts de mostreig. És adequat per a superfícies que varien lleument. La superfície ha de passar exactament pels punts de dades i ha de tenir una curvatura mínima. [16]

1.4.6. Spline amb barreres

La Spline amb barreres utilitza un mètode similar a spline, però aquesta distingeix les discontinuïtats codificades tant en les barreres d'entrada com en les dades del punt d'entrada. Aplica un mètode de curvatura mínima, a través d'una tècnica multi-quadrícula unidireccional, que va d'una quadrícula inicial gruixuda, inicialitzada com la mitjana de les dades d'entrada a través d'una sèrie de quadrícules més fines fins a produir una aproximació d'una superfície de curvatura mínima en l'espaiat de fila i columna desitjades.

1.4.7. Topo to raster

Per a calcular amb *topo to raster* s'utilitza una tècnica d'interpolació específicament per a crear una superfície que representa amb més precisió una superfície natural de drenatge i preserva millor les cordes muntanyoses i les xarxes de transmissió de les dades de corbes de nivell d'entrada. L'algoritme utilitzat està basant en el de ANUDEM desenvolupat a la Universitat Nacional d'Austràlia per Hutchinson i altres.

Interpola els valors d'elevació per a un raster mentre que s'imposen restriccions que assegurin una estructura de drenatge connectat i la correcta representació de crestes i zones enfonsades a partir de les dades de corbes de nivell d'entrada. [2]

1.4.8. Tendència

La tendència és una interpolació polinòmica global que ajusta una superfície suau definida per una funció matemàtica als punts de mostra d'entrada. Canvia gradualment i captura patrons d'escala sense detallar les dades.

Aquest tipus d'interpolació es pot utilitzar per a ajustar una superfície als punts de mostra quan la superfície varia gradualment d'una regió a una altra sobre l'àrea d'interès. [2]

1.4.9. Comparació kriging amb altres mètodes

Pot haver-hi millors estimacions mitjançant mètodes no lineals o esbiaixats, ja que el compliment de les suposicions es crític i en cas d'utilitzar un semivariograma inadequat no es poden garantir les propietats encara que la interpolació sigui amb resultats aparentment bons, i que en el cas de que no hi hagi dependència espacial, la interpolació amb kriging no està justificada. [4] Però és el mètode més adequat quan se sap que hi ha influència direccional o de la distància correlacionada espacialment a les dades.

El mètode està relacionat matemàticament amb l'anàlisi de regressió. Les dues teories deriven en un millor estimador lineal no esbiaixat, basat en suposicions o covariàncies, utilitzant el teorema de Gauss-Markov per a comprovar la independència de l'estimat i l'error, i fan ús d'una fórmula molt similar. Per tant, són molt útils en diferents camps de treball, el kriging es fa per a estimar una sola realització d'un camp aleatori mentre que els models de regressió es basen en múltiples observacions de dades multivariades.

La interpolació per kriging pot ser vista com una spline en un espai de reproducció Hilbert Kernel (RKHS, *Reproducing Kernel Hilbert space*). La diferència ve donada per la interpretació: mentre que la spline està motivada per la mínima norma d'interpolació basada en una estructura d'espai de Hilbert, el kriging està motivat per una predicció de l'error al quadrat, basat en un model estocàstic.

El kriging amb una superfície de tendències polinòmiques és matemàticament idèntic a la corba polinòmica de muntatge de mínims quadrats generalitzats.

També pot ésser entès com una forma d'inferència bayesiana, que és un mètode d'inferència estadística important en l'anàlisi dinàmic d'una seqüència de dades, en la qual el teorema de Bayes s'utilitza per a actualitzar la probabilitat d'una hipòtesi com a prova adquirida.

Les eines d'interpolació IDW i spline són considerades mètodes d'interpolació determinístics perquè estan basats directament en els valors mesurats circumdants o en fórmules matemàtiques especificades que determinen la suavitat de la superfície resultant. Hi ha una segona família de mètodes d'interpolació que consta de mètodes geoestadístics, com el kriging que està basat en models estadístics que inclouen la autocorrelació, és a dir, les relacions estadístiques entre els punts mesurats. Les tècniques d'estadística geogràfica no només tenen capacitat de produir una superfície de predicció sinó que també proporcionen alguna mesura de certesa o precisió de les prediccions.

CAPÍTOL 2. MODELAT DE DADES

Es vol crear un model matemàtic d'interpolació de dades aeronàutiques a partir de certs valors coneguts. Per fer-ho possible serà necessària la construcció d'un programa que sigui capaç de processar les dades d'entrada conegudes per a fer una estimació dels valors no coneguts mitjançant el mètode de kriging ordinari, utilitzant els semivariogrames esfèric, exponencial i gaussià, ja que són els més comuns i els que més s'utilitzen.

2.1. Anàlisi de dades

Les dades utilitzades per a realitzar els tests principals d'aquest treball són variables geomètriques d'un component d'una turbina d'alta pressió d'un avió, on volem obtenir el *tip clearance*, que és la distància entre la punta de les pales de la turbina i la carcassa exterior.

Aquest paràmetre ha de ser el més petit possible per millorar l'eficiència del motor, però prou gran com per evitar col·lisions fins i tot durant la rotació del motor a alta velocitat i tenint en compte la dilatació corresponent dels materials. [17]

No es donaran dades del fabricant ni del projecte per temes de confidencialitat. S'han utilitzat dos fitxers de dades conegudes per a interpolar els valors de *tip clearance* desitjats. En el primer fitxer només es tenen en compte dos paràmetres i en el segon hi ha tres paràmetres més a tenir en compte a més a més del dos anteriors.

2.1.1. Dos paràmetres d'entrada

Hi ha dos paràmetres d'entrada (x_1 i x_2) i una variable de sortida, que és el *tip clearance*, el valor d'interès, per a cadascuna de les dades conegudes.

La primera variable correspon a la posició respecte a la circumferència de l'anclatge del motor, que està entre -15° i 15° , tal i com es pot veure a la **Fig 2.1**.

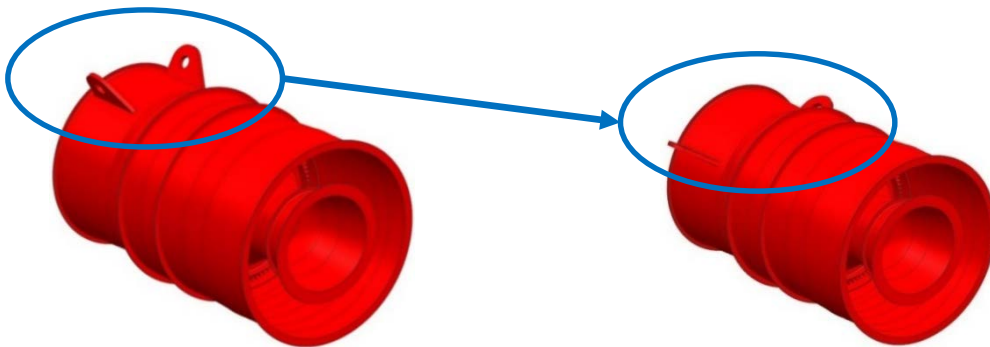


Fig 2.1 Variació de la posició en la circumferència de l'anclatge del motor

La segona variable correspon a l'angle en el que està col·locat l'anclatge i va entre -15 i 15° com es pot veure a la **Fig 2.2**.

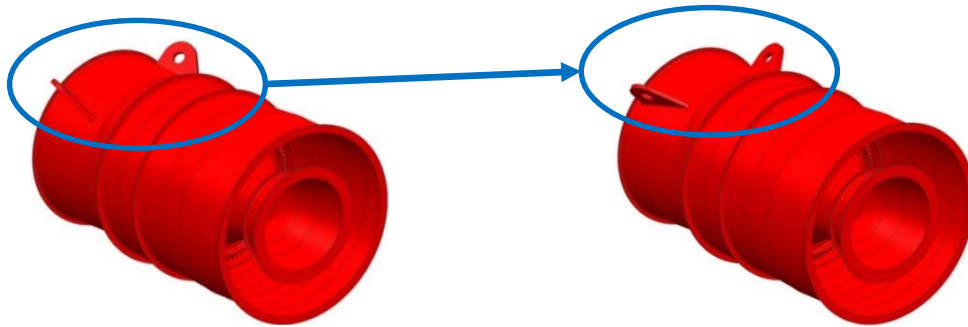


Fig 2.2 Variació de l'angle de l'anclatge del motor

2.1.2. Cinc paràmetres d'entrada

En el cas dels cinc paràmetres d'entrada (x_1 , x_2 , x_3 , x_4 i x_5) té la mateixa variable de sortida. A més dels dos paràmetres indicats a l'apartat 272.1.1. hi ha tres més.

La tercera variable correspon al gruix de la carcassa del motor, que varia entre -2 i 1 mil·límetres, com es pot veure a la **Fig 2.3**



Fig 2.3 Gruix de la carcassa del motor

La quarta variable és la posició axial de l'anell de rigidesa addicional que es troba a la carcassa, que s'observa a la **Fig 2.4**

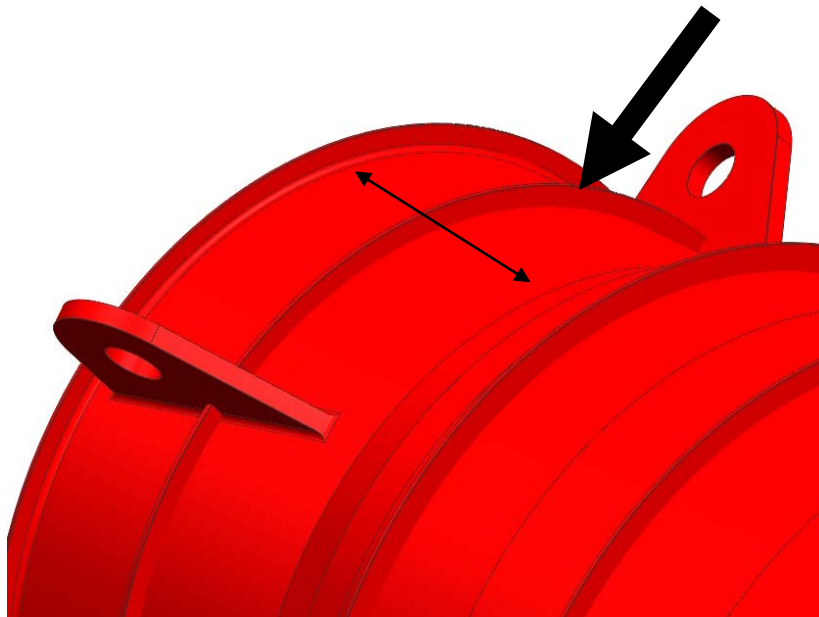


Fig 2.4 Posició de l'anell de rigidesa addicional

I la cinquena i última variable que s'analitza en aquests punts és el gruix de l'anell de rigidesa addicional del que es parla a la variable anterior, que es troba entre 2 i 4 mil·límetres, tal i com es veu a la **Fig 2.5**

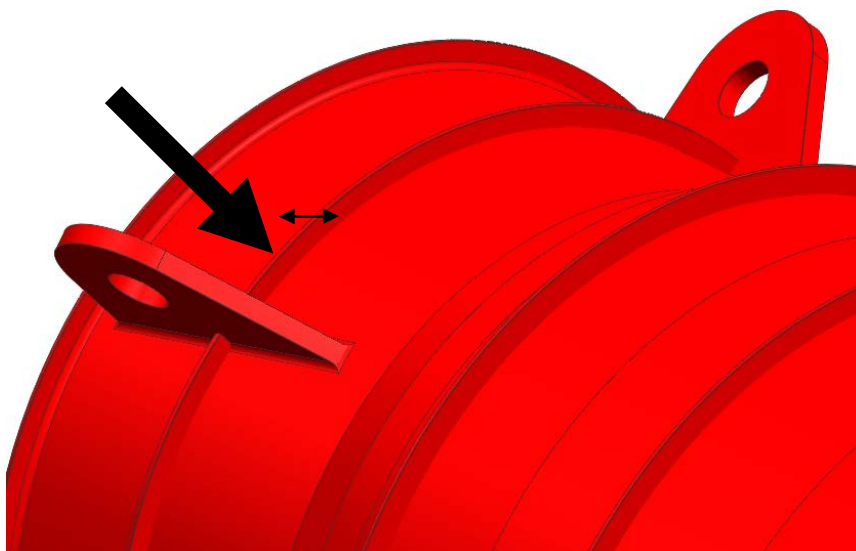


Fig 2.5 Gruix de l'anell de rigidesa addicional

2.2. Elecció del model i disseny del programa

Després d'analitzar els avantatges de treballar amb l'estimació per kriging vists al CAPÍTOL 1 i d'analitzar el tipus de dades que es volen interpolar, es considera el mètode més adient per a fer els càlculs d'interpolació d'aquestes dades, ja que és un sistema amb molta precisió i que indica l'error dels càlculs realitzats.

Com que s'han de gestionar moltes dades, al principi s'agafaven només els set punts més propers al punt que s'està interpolant, el que es vol conèixer, ja que el càlcul del determinant de les matrius per a calcular la inversa es feia impossible de gestionar, si s'afegien més punts el temps s'incrementava de manera exponencial, i encara que era capaç de fer el càlcul, trigava hores o fins i tot dies. Aquesta solució es va buscar perquè en la bibliografia consultada es va veure que era un recurs bastant utilitzat, agafar només els punts propers que són els que realment afecten significativament al càlcul i ignorar els que són més lluny.

Després es va afegir una llibreria, *Apache Commons Math*, que fa el càlcul de la inversa de manera òptima, utilitza el mètode de la inversa generalitzada de Moore-Penrose. El que fa es descomposar la matriu en altres més simples **(2.1)** i, utilitzant les matrius transposades conjugades d'aquestes (que si són matrius de nombres reals són iguals que les matrius transposades) **(2.2)**, calcula la pseudoinversa, que si la matriu és invertible, la pseudoinversa és el mateix que la inversa **(2.3)** [18].

$$A = BC \rightarrow A_{m \times n}, B_{m \times r}, C_{r \times n} \quad (2.1)$$

$$A^+ = C^*(CC^*)^{-1}(B^*B)^{-1}B^* \quad (2.2)$$

$$A^+ = A^{-1} \quad (2.3)$$

Això permet que es pugui treballar amb tots els punts introduïts, el resultat encara que no varia tant, perquè com ja s'ha dit, els punts llunyans no afecten gaire al resultat, és cert que la precisió augmenta quan hi ha més punts coneguts de referència i com que s'està treballant amb toleràncies, és a dir, amb mides molt petites, és important que hi hagi la màxima precisió possible.

El programa s'ha dissenyat en el llenguatge Java perquè es va considerar adequat gràcies al seu tipus de programació i perquè es tenien més coneixements previs que d'altres llenguatges similars, cosa que facilitava molt la feina que calia fer. Com que és un llenguatge orientat a objectes, és molt escalable, així que si algú vol ampliar o modificar el codi en un futur és molt més fàcil que en altres tipus de llenguatges de programació. Per a executar-se és independent de la plataforma, ja que s'executa en una màquina virtual anomenada JVM, així que serà compatible amb qualsevol plataforma amb aquest simulador, és a dir no caldrà un tipus de màquina específica per a fer anar el programa, tindrà més compatibilitat.

El primer que es necessita per a treballar amb kriging són les dades conegudes, per això el programa, (veure **Fig 2.6**) llegeix des d'un fitxer de text, un arxiu seleccionat per l'usuari on les dades de la primera columna corresponen a x_1 , la segona a x_2 i així successivament fins a arribar a x_n , la última columna correspon als *tip clearance* coneguts per als paràmetres mesurats, és a dir, els valors que són d'interès per a aquest estudi.

El programa és capaç de realitzar els càlculs des de dades d'una sola dimensió, és a dir, una recta amb valors per a cada punt, fins a dades de les dimensions que es desitgi, però s'ha de tenir en compte que si les dades tenen molts paràmetres que es tenen en compte per als resultats, el temps per a realitzar el càlcul augmentarà exponencialment, perquè les possibilitats que ha de calcular a l'hora de buscar les distàncies seran moltes més si els punts tenen més dimensions.

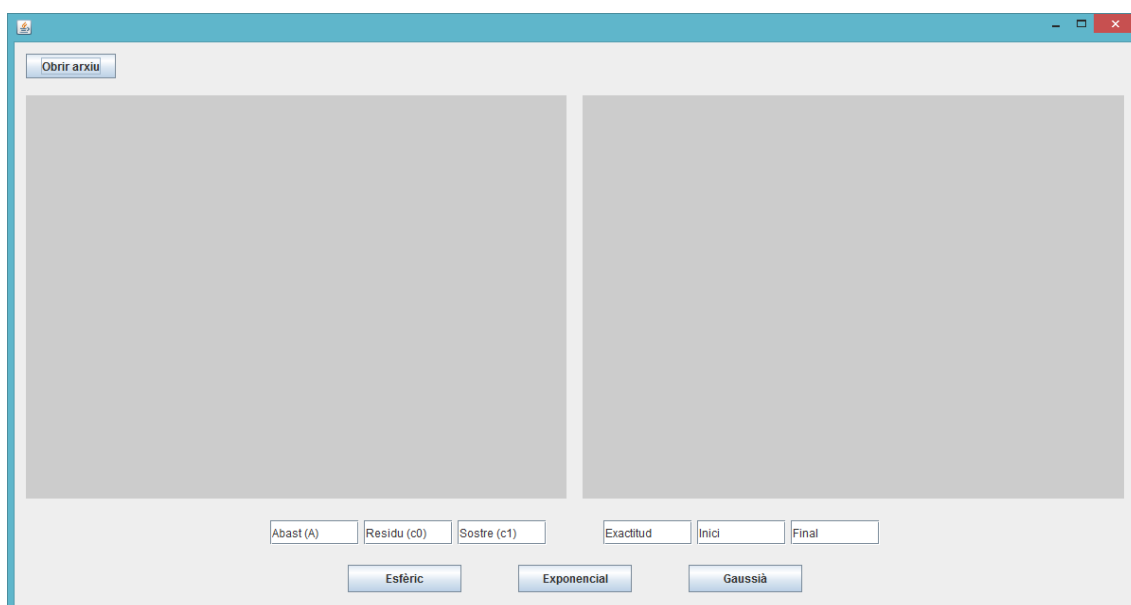


Fig 2.6 Visualització del programa

El programa necessita que les columnes del fitxer d'entrada estiguin separades per punt i coma i els decimals estiguin indicats mitjançant un punt. Com es pot veure a la **Fig 2.7**.

Es va escollir aquest format per llegir les dades d'entrada ja que al Microsoft Excel hi ha una opció per exportar els arxius de dades separades en columnes on es pot seleccionar aquest format, que és el .CSV, i es va considerar que si es tenien fitxers amb grans quantitats de dades seria molt fàcil transformar-les utilitzant aquesta opció disponible.

```

0.2625;0.2125;0.010710273
0.1125;0.1625;0.011375666
0.3125;0.2875;0.010553536
0.3375;0.0375;0.010727522
0.0375;0.5125;0.011634609
0.2875;0.9125;0.010399201
0.8125;0.6375;0.010379186
0.5125;0.4375;0.010399045
0.6375;0.2375;0.010509453

```

Fig 2.7 Exemple d'un fitxer de text de dades 2D

Un cop el programa ha llegit el fitxer d'entrada de dades, els valors de les distàncies i les seves semivariàncies les col·loca en un gràfic de punts on l'eix X és la distància i l'eix Y la semivariància calculada. Aquest gràfic s'anomena semivariograma empíric. Es pot veure a la **Fig 2.8**.

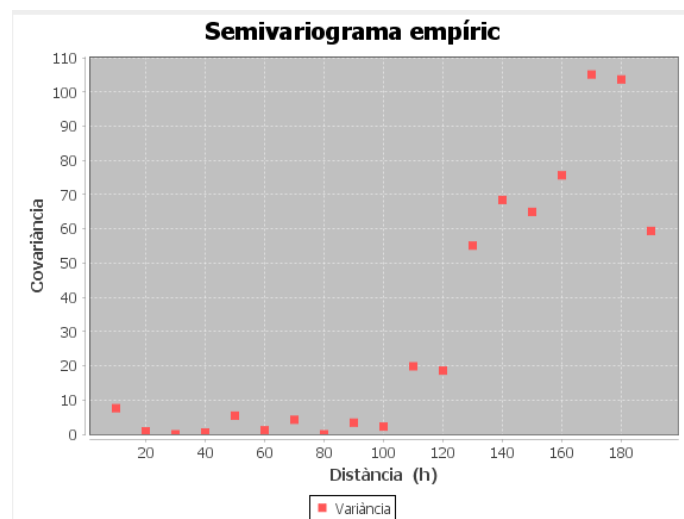


Fig 2.8 Exemple del semivariograma empíric calculat i mostrat al programa

Quan ja s'ha obtingut el semivariograma empíric l'usuari ha d'omplir les dades necessàries per al següent càlcul, que són: l'abast, el sostre i el residu que s'han detallat a l'apartat 1.3.2. Com una ajuda addicional, un cop s'han introduït aquestes dades, el programa pinta els tres tipus de semivariograma més comuns, que són els que es podran utilitzar. Si es varien les dades, el dibuix també variarà. Aquest dibuix serveix per comprovar visualment si l'abast, sostre i residu introduïts són els adequats per a les dades conegudes i fer els ajusts que es cregui convenient. Es pot veure un exemple de les gràfiques dibuixades a la **Fig 2.9** on el vermell és l'esfèric, el blau és l'exponencial i el verd és el gaussià.

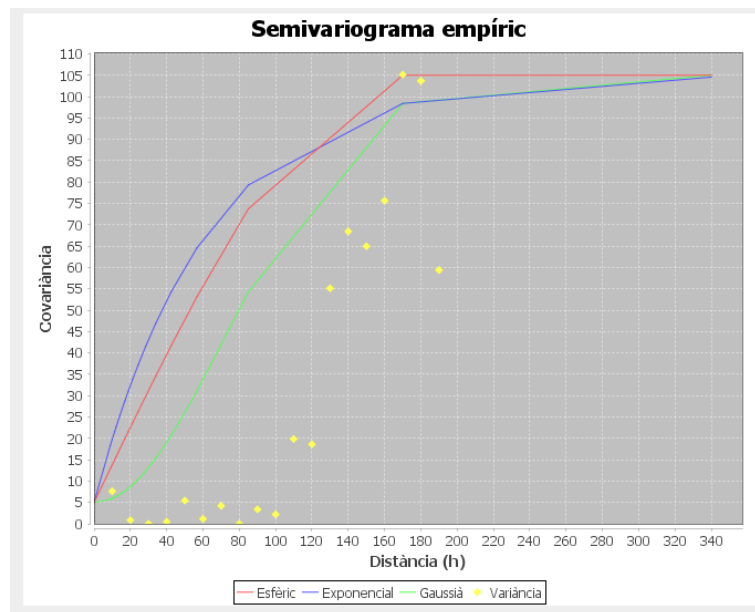


Fig 2.9 Exemple de semivariogrames mostrats al programa

Quan l'usuari decideix que les dades introduïdes són correctes ja pot omplir l'exactitud que vol, és a dir, quants punts per tram s'interpolaran, i el rang on es realitzaran els càlculs. Demana només un inici i un final, ja que agafa tots els eixos amb el mateix inici i el mateix final, és a dir, una zona quadrada, cúbica, etc., segons les dimensions que hi hagi. Després l'usuari ha d'escollir entre Esfèric, Exponencial o Gaussià per seleccionar quin del models de semivariograma es vol utilitzar per a la realització dels càlculs, és a dir, quin és el que més s'ajusta als punts mostrats. Quan el programa realitzi els càlculs del kriging explicats a l'apartat 1.2.4.3, els mostrarà en forma de superfície de colors com es pot veure a la **Fig 2.10**, només disponible per a càlculs en tres dimensions, ja que amb més dimensions no es podria representar d'aquesta manera i amb menys no tindria massa sentit fer-ho així.

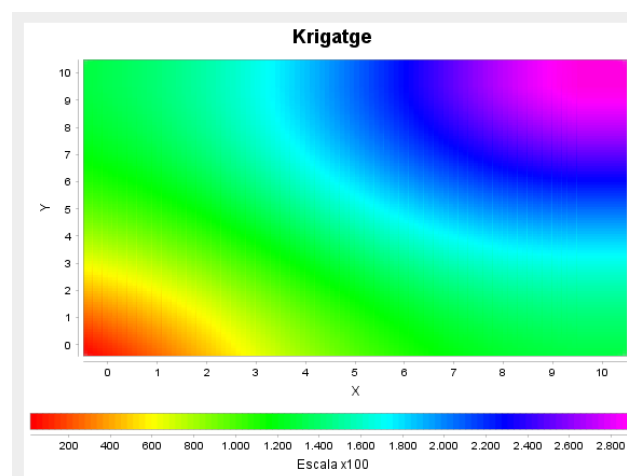
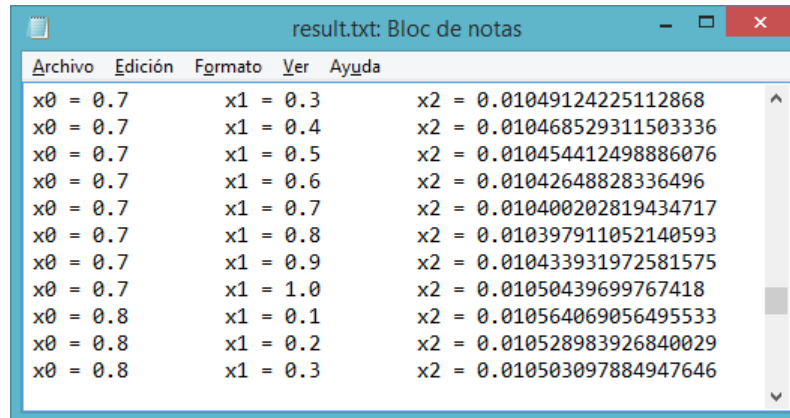


Fig 2.10 Exemple de mapa de colors d'un càlcul per kriging amb s. Gaussià

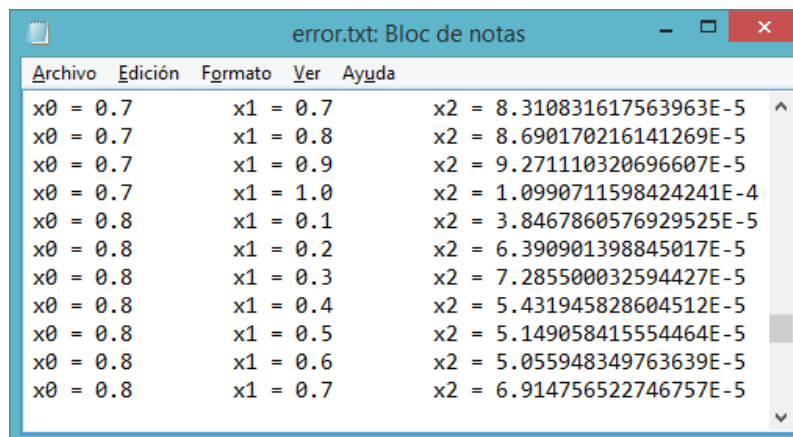
Si es creen pocs punts per al rang seleccionat, de manera que no s'omple tot l'espai del gràfic mostrat, no es veurà continu com a la **Fig 2.10**, es veuran punts separats.

Un cop finalitzat el procés de càlcul, es poden trobar a la mateixa carpeta on s'ha executat el programa els arxius de text result.txt i error.txt, que són els resultats obtinguts amb la interpolació i l'error de kriging per als càlculs. Seran arxius similars als mostrats a les **Fig 2.11** i **Fig 2.12**.



Archivo	Edición	Formato	Ver	Ayuda
x0 = 0.7	x1 = 0.3	x2 = 0.01049124225112868		
x0 = 0.7	x1 = 0.4	x2 = 0.010468529311503336		
x0 = 0.7	x1 = 0.5	x2 = 0.010454412498886076		
x0 = 0.7	x1 = 0.6	x2 = 0.01042648828336496		
x0 = 0.7	x1 = 0.7	x2 = 0.010400202819434717		
x0 = 0.7	x1 = 0.8	x2 = 0.010397911052140593		
x0 = 0.7	x1 = 0.9	x2 = 0.010433931972581575		
x0 = 0.7	x1 = 1.0	x2 = 0.01050439699767418		
x0 = 0.8	x1 = 0.1	x2 = 0.010564069056495533		
x0 = 0.8	x1 = 0.2	x2 = 0.010528983926840029		
x0 = 0.8	x1 = 0.3	x2 = 0.010503097884947646		

Fig 2.11 Exemple de result.txt



Archivo	Edición	Formato	Ver	Ayuda
x0 = 0.7	x1 = 0.7	x2 = 8.310831617563963E-5		
x0 = 0.7	x1 = 0.8	x2 = 8.690170216141269E-5		
x0 = 0.7	x1 = 0.9	x2 = 9.271110320696607E-5		
x0 = 0.7	x1 = 1.0	x2 = 1.0990711598424241E-4		
x0 = 0.8	x1 = 0.1	x2 = 3.8467860576929525E-5		
x0 = 0.8	x1 = 0.2	x2 = 6.390901398845017E-5		
x0 = 0.8	x1 = 0.3	x2 = 7.285500032594427E-5		
x0 = 0.8	x1 = 0.4	x2 = 5.431945828604512E-5		
x0 = 0.8	x1 = 0.5	x2 = 5.149058415554464E-5		
x0 = 0.8	x1 = 0.6	x2 = 5.055948349763639E-5		
x0 = 0.8	x1 = 0.7	x2 = 6.914756522746757E-5		

Fig 2.12 Exemple de error.txt

En aquests arxius la primera columna correspon a les dades del primer paràmetre, la segona a les del segon, la tercera a les del tercer i així successivament fins a arribar a la última columna que és la que correspon al resultat per a result.txt i a l'error per a error.txt.

2.3. Processos interns del programa

El programa a més de mostrar una sèrie de gràfiques i de resultats també realitza els càlculs necessaris de manera interna, no tots són visibles per l'usuari. Els passos que segueix són els següents:

- 1) Quan l'usuari selecciona l'arxiu de text, el comença a llegir i per a cada dada llegida crea un objecte de la "classe Coordenadas". Quan ha llegit una línia, afegeix les seves "Coordenadas" a un objecte de la classe "Punto", i afegeix cada "Punto" a una "llista de Puntos".
- 2) Un cop ha acabat de llegir l'arxiu, es canvia una variable booleana de fals a cert, que indica que l'arxiu està tot llegit i que no ha de continuar posant més punts a la llista.
- 3) Inicialitza tots els objectes que ha d'utilitzar per a la resta de processos perquè la execució anterior, en el cas que ja s'hagi realitzat una, no afecti als nous resultats.
- 4) Calcula totes les distàncies entre tots els parells de punts d'entrada, les ordena i calcula la covariància per a cada distància existent, realitzant una mitjana en el cas que una distància estigui repetida.
- 5) Per a cada punt de la covariància crea un objecte de tipus "Punto" amb les seves "Coordenadas" corresponents i els posa a una "llista de Puntos", igual que feia amb els punts d'entrada. Crida a una funció que pinta tots aquests punts a la primera gràfica. Tal i com es veu a la **Fig 2.8**.
- 6) El programa principal es troba en un bucle constant esperant que s'activin algunes variables per a poder continuar. Les variables són:
 - a. Que hi hagi un arxiu llegible amb punts, que s'ordenin les distàncies, calculi la covariància i es dibuixi a la primera gràfica.
 - b. Que s'activi una variable que activa la classe "KrigingFrame" (que és la part gràfica del programa) quan detecta valors vàlids en els tres primers camps. Aleshores comença la funció que els agafa i els compara amb els anteriors i si són diferents, en cas que no hi hagi anteriors sempre seran diferents, (aquesta part de comparació es fa perquè pari d'actualitzar la gràfica i permeti fer zoom, perquè sinó no pararia de pintar el mateix tota l'estona), pinta a la primera gràfica les equacions de covariància esfèrica, exponencial i gauss amb les dades introduïdes, des del punt zero fins al doble del rang, aquest segment el divideix en vint parts que seran els punts de l'eix X, i per a aquests punts calcula la Y amb cada funció corresponent. El resultat d'aquesta part és el que es pot observar a la **Fig 2.9**.

- c. La tercera variable s'activa quan estan tots els camps plens amb valors vàlids i l'usuari prem un dels tres botons (Esfèric, Exponencial o Gaussià). Aleshores llegeix els paràmetres introduïts, i per fer-ho, "KrigingFrame" crea un nou objecte de la classe "Parametros", el llegeix, s'inicialitzen altres classes que afecten només per a pintar la segona gràfica (per si de cas es prem un altre botó que també ho pinti) i es creen les "Coordenadas" finals agafant els valors introduïts amb una funció recursiva en la que crea els nous "Puntos" amb el que s'ha introduït.
- 7) Es calcula la variància i s'omple una matriu, es fa la inversa amb la llibreria esmentada al capítol anterior.
- 8) Fa un bucle per a tots els punts finals que s'han creat i per a cadascun executa la funció de Kriging on es realitza el càlcul de la distància dels punts finals creats amb cadascun dels punts d'entrada, calcula el pes i el resultat, es pot veure a la **Fig 2.13**.

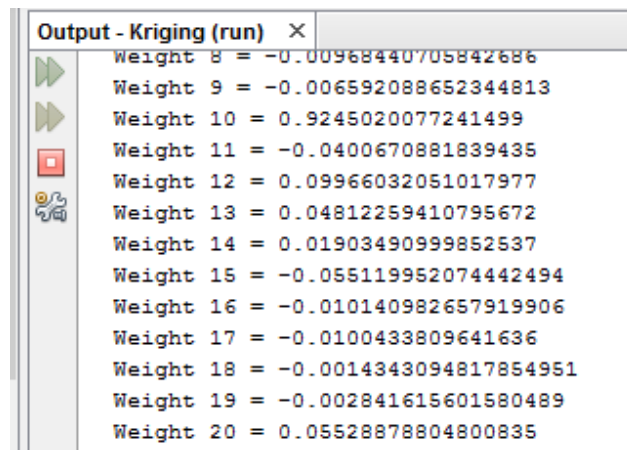


Fig 2.13 Exemple de la llista de pesos

- 9) El "Resultado" crea un punt amb les coordenades del punt creat i la última coordenada serà el resultat, fa una llista d'aquests punts i quan detecta que està plena, només si es amb tres paràmetres, els pinta a la segona gràfica (**Fig 2.10**), i sempre escriu el resultat en un fitxer amb la llista de punts. (**Fig 2.11**)
- 10) Fa el mateix procés amb l'error, el calcula i l'introdueix en un fitxer amb la llista de punts. (**Fig 2.12**)
- 11) Si es canvien les dades introduïdes per l'usuari (o no), sense canviar de fitxer d'entrada i es selecciona un altre model de semivariograma es torna a executar des del pas 7 i canvien els resultats i la segona gràfica com es veu a la **Fig 2.14**, si es canvia de fitxer es comença des de l'inici.

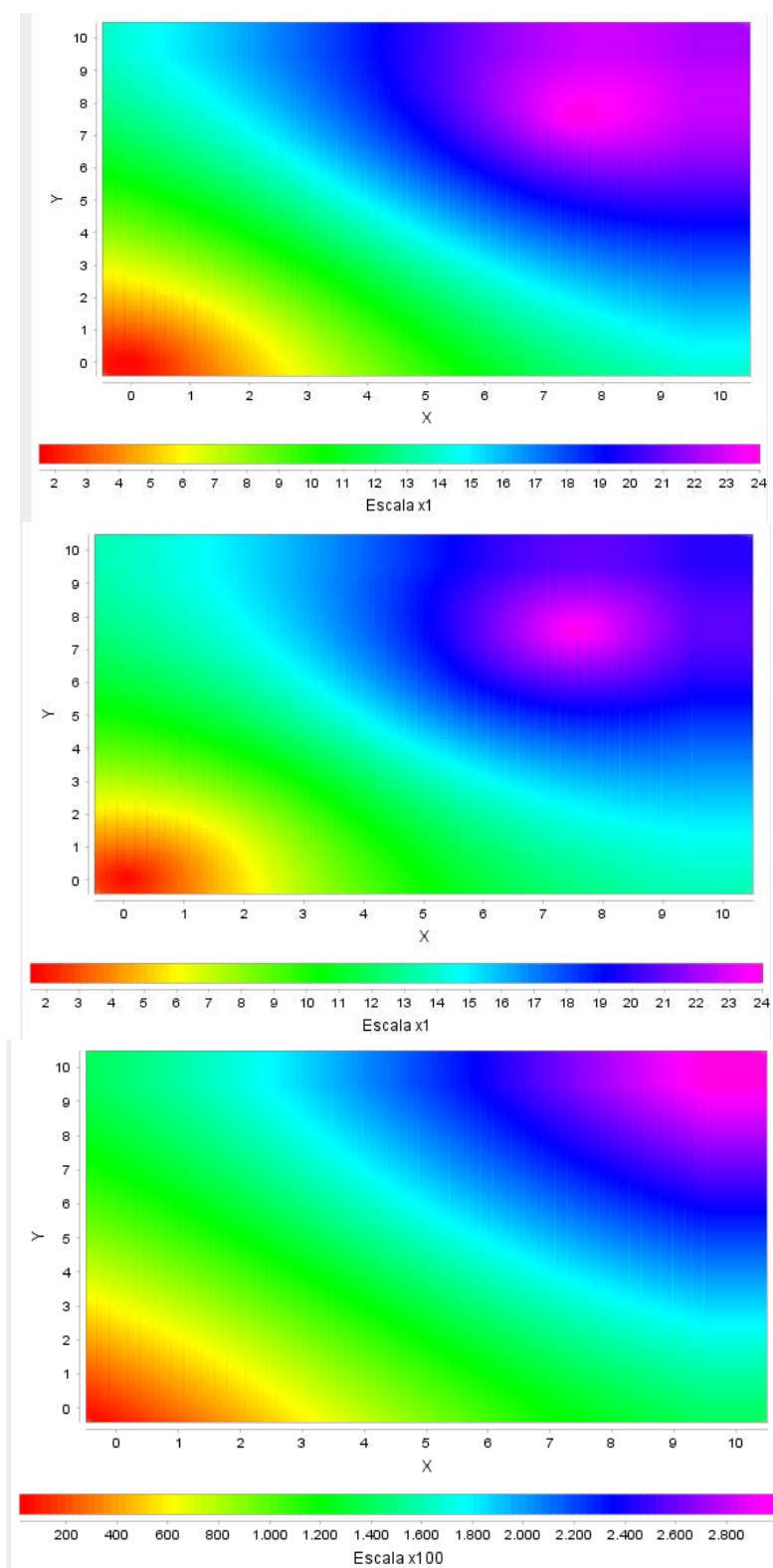


Fig 2.14 Canvis en model de semivariograma sense variar les dades ni el fitxer

CAPÍTOL 3. RESULTATS OBTINGUTS

3.1. Comprovació del funcionament

Per comprovar que el programa realitza els càlculs de manera correcta i que està fent tot el desitjat, a més de fer una sèrie de càlculs molt senzills a mà i comprovar-los, s'han agafat molts exemples de càlcul d'interpolació per kriging ja realitzats prèviament per a altres camps on s'utilitza molt, per exemple la geoestadística, la mineria i la medicina i s'ha comprovat que la resposta de sortida sigui similar als exemples i al programa, per tal d'assegurar-se que faci el procediment de manera correcta.

En aquest document s'inclouen els exemples més destacats dels models que s'han comprovat. Ja que la resta no eren dades en 2 dimensions i visualment no es pot comparar, es van comparar les dades numèriques dels resultats.

Es va trobar un arxiu que acompanyava al programa GeoEas on hi havia una sèrie de dades sobre el nivell de cadmi en una zona, amb 60 blocs de dades conegudes, que indicaven coordenada X i Y al terreny estudiat i la quantitat de cadmi en aquest punt [19]. Es va realitzar el càlcul de la interpolació per kriging i els resultats obtinguts eren similars als càlculs realitzats a l'informe de referència, com es pot comprovar a les **Fig 3.1** i **Fig 3.2**.

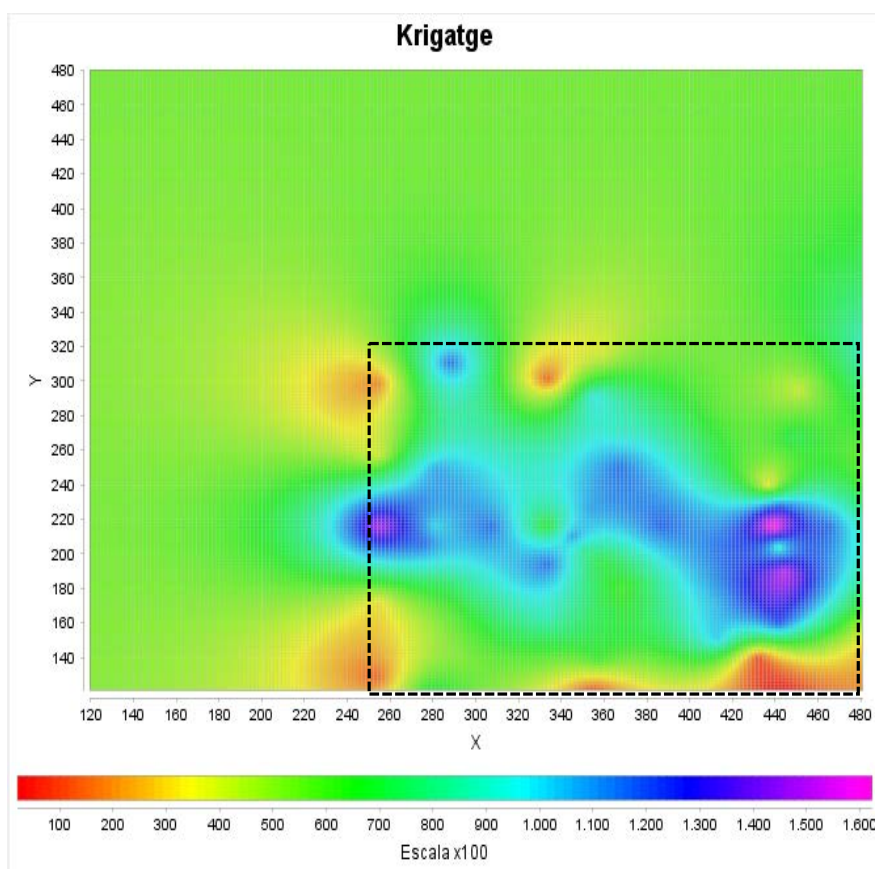


Fig 3.1 Interpolació amb el programa d'estudi dels nivells de cadmi

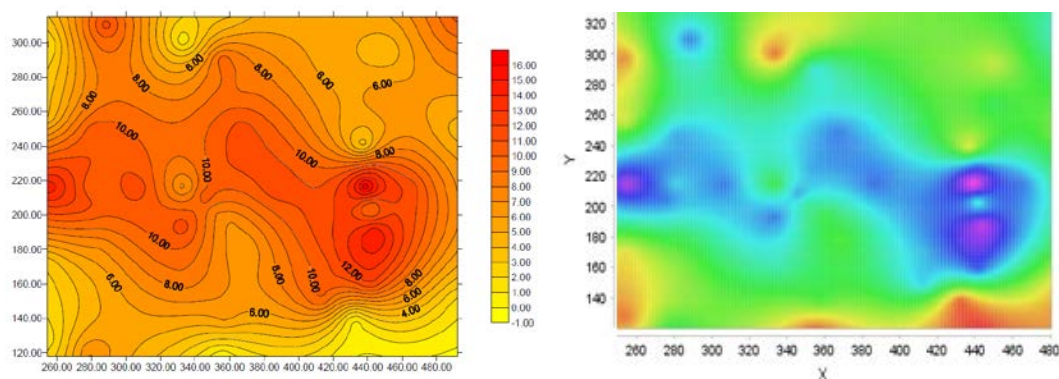


Fig 3.2 Comparació amb la interpolació de P. M. Barbosa

Un altre conjunt de dades trobat va ser un anàlisi de la contaminació nuclear de l'aqüífer conegut com a Wolfcamp situat a Texas, Estats Units. [20] Es tenen 80 punts de referència, on s'indiquen les coordenades on es va mesurar cada dada i el cap piezomètric mesurat per sobre del nivell del mar. També es va comprovar la interpolació i els resultats es poden veure a les **Fig 3.3** i **Fig 3.4**.

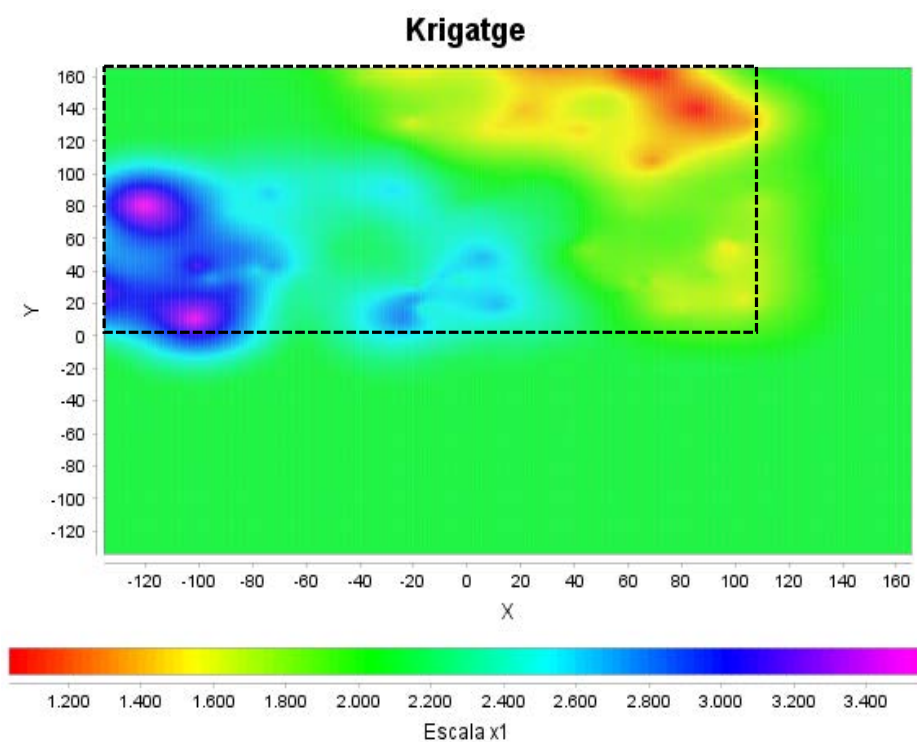


Fig 3.3 Interpolació amb el programa de la contaminació de l'aqüífer

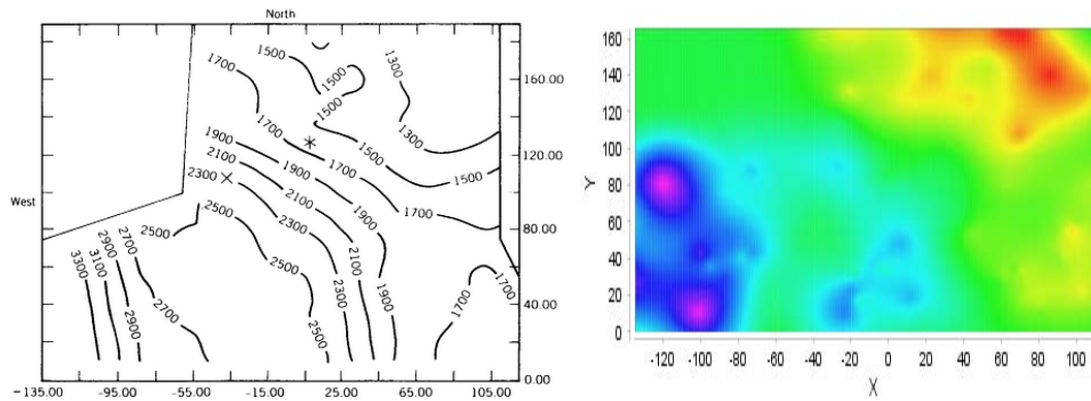


Fig 3.4 Comparació dels resultats obtinguts amb els de N. Cressie

3.1.1. Rang de càlcul correcte dels resultats

S'ha comprovat que si es vol realitzar una interpolació a distàncies llunyanes dels punts coneguts d'entrada, com que els pesos deixen d'afectar als càlculs i el model de semivariograma es torna constant, no són resultats fiables, ja que són resultats constants per a tot l'espai.

Així que només es pot treballar en la zona de les mostres de les dades conegudes per a fer les interpolacions. Es pot veure a la **Fig 3.5** on a partir d'un cert número els valors del resultat són constants, en el semivariograma esfèric, o la seva variació és mínima en al cas dels semivariogrames exponencial i gaussià.

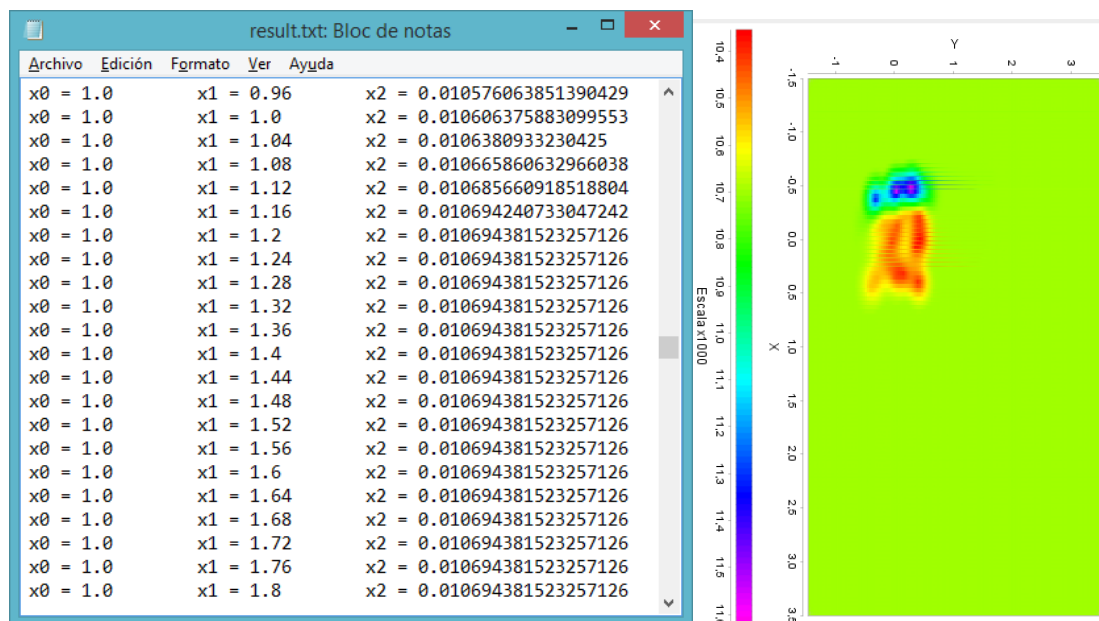


Fig 3.5 Interpolació per comprovar el rang en s. esfèric

3.2. Simulacions del *tip clearance*

S'han realitzat una sèrie de simulacions amb els mateixos valors coneguts d'entrada però variant el tipus de semivariograma i les dades d'abast, sostre i residu i variant també la precisió, per poder ajustar-se millor als resultats desitjats.

Com s'ha comprovat a l'apartat 3.1.1 utilitzarem el rang des de 0 que seria la mínima distància possible fins a 1, ja que si es volen obtenir resultats per a un rang que el màxim sigui més d'1, s'haurien de tenir dades conegudes situades a la zona que es vulgui saber.

3.2.1. Simulacions amb les dades 2D

Per la les interpolacions de dades de 2 paràmetres s'han realitzat càlculs agafant l'arxiu de dades conegudes de 2D i s'ha comprovat, amb una exactitud de 100, és a dir per a 10100 punts, quan hi ha l'error més baix. S'han realitzat diverses proves, fins a trobar els paràmetres de rang, sostre i residu més adequats i el model adient per a aquestes dades. Una selecció dels més significatius d'aquests càlculs realitzats es pot observar a la **Taula 3.1**.

Taula 3.1 Resultats simulacions 2D

	R min	R màx	R mig	Error min	Error màx	Error mig
a	$1,03428 \cdot 10^{-2}$	$1,16713 \cdot 10^{-2}$	$1,06255 \cdot 10^{-2}$	$1,12553 \cdot 10^{-4}$	$7,57585 \cdot 10^{-4}$	$4,38609 \cdot 10^{-4}$
b	$1,03352 \cdot 10^{-2}$	$1,16817 \cdot 10^{-2}$	$1,06234 \cdot 10^{-2}$	$8,29158 \cdot 10^{-5}$	$6,47219 \cdot 10^{-4}$	$3,32550 \cdot 10^{-4}$
c	$1,03428 \cdot 10^{-2}$	$1,16713 \cdot 10^{-2}$	$1,06255 \cdot 10^{-2}$	$3,75177 \cdot 10^{-5}$	$2,52528 \cdot 10^{-4}$	$1,46203 \cdot 10^{-4}$
d	$1,03352 \cdot 10^{-2}$	$1,16817 \cdot 10^{-2}$	$1,06234 \cdot 10^{-2}$	$2,93920 \cdot 10^{-5}$	$2,15971 \cdot 10^{-4}$	$1,11331 \cdot 10^{-4}$
e	$1,03678 \cdot 10^{-2}$	$1,16222 \cdot 10^{-2}$	$1,06655 \cdot 10^{-2}$	$3,24473 \cdot 10^{-5}$	$1,03645 \cdot 10^{-4}$	$9,88753 \cdot 10^{-5}$
f	$1,02937 \cdot 10^{-2}$	$1,16730 \cdot 10^{-2}$	$1,06212 \cdot 10^{-2}$	$1,19929 \cdot 10^{-5}$	$8,44075 \cdot 10^{-5}$	$4,77725 \cdot 10^{-5}$
g	$1,03352 \cdot 10^{-2}$	$1,16817 \cdot 10^{-2}$	$1,06234 \cdot 10^{-2}$	$8,74009 \cdot 10^{-7}$	$6,82229 \cdot 10^{-6}$	$3,50538 \cdot 10^{-6}$
h	$1,03368 \cdot 10^{-2}$	$1,16832 \cdot 10^{-2}$	$1,06221 \cdot 10^{-2}$	$7,37000 \cdot 10^{-7}$	$6,02356 \cdot 10^{-6}$	$2,97720 \cdot 10^{-6}$

Taula 3.2 Paràmetres simulacions taula 3.1

	Abast	Sostre	Residu	Model		Abast	Sostre	Residu	Model
a	1,3	$9 \cdot 10^{-7}$	0	Expon.	e	0,1	10^{-9}	0	Esfèric
b	1,3	$9 \cdot 10^{-7}$	0	Esfèric	f	0,7	10^{-8}	0	Esfèric
c	1,3	$1 \cdot 10^{-7}$	$1 \cdot 10^{-10}$	Expon.	g	1,3	10^{-10}	0	Esfèric
d	1,3	$1 \cdot 10^{-7}$	$1 \cdot 10^{-10}$	Esfèric	h	1,8	10^{-10}	0	Esfèric

3.2.2. Simulacions amb les dades 5D

Per a les interpolacions de dades de 5 paràmetres s'han realitzat diverses simulacions amb l'arxiu 5D, agafant una exactitud de 5, és a dir, 22500 punts, s'ha fet una selecció dels resultats més significatius obtinguts i es pot veure a la **Taula 3.3**. S'han fet diverses simulacions fins a trobar els paràmetres òptims.

Taula 3.3 Resultats simulacions 5D

	R min	R màx	R mig	Error min	Error màx	Error mig
j	$9,97133 \cdot 10^{-3}$	$1,15639 \cdot 10^{-2}$	$1,04851 \cdot 10^{-2}$	$1,10777 \cdot 10^{-4}$	$2,94045 \cdot 10^{-4}$	$2,21834 \cdot 10^{-4}$
k	$9,90658 \cdot 10^{-3}$	$1,16689 \cdot 10^{-2}$	$1,04988 \cdot 10^{-2}$	$8,54307 \cdot 10^{-5}$	$2,68781 \cdot 10^{-4}$	$1,78174 \cdot 10^{-4}$
l	$9,90658 \cdot 10^{-3}$	$1,16689 \cdot 10^{-2}$	$1,04988 \cdot 10^{-2}$	$4,14710 \cdot 10^{-5}$	$9,06328 \cdot 10^{-5}$	$6,46641 \cdot 10^{-5}$
m	$9,90658 \cdot 10^{-3}$	$1,16689 \cdot 10^{-2}$	$1,04988 \cdot 10^{-2}$	$2,68298 \cdot 10^{-5}$	$8,49371 \cdot 10^{-5}$	$5,62528 \cdot 10^{-5}$
n	$9,91361 \cdot 10^{-3}$	$1,16659 \cdot 10^{-2}$	$1,04863 \cdot 10^{-2}$	$2,00242 \cdot 10^{-5}$	$6,54680 \cdot 10^{-5}$	$4,10690 \cdot 10^{-5}$
o	$9,91692 \cdot 10^{-3}$	$1,16658 \cdot 10^{-2}$	$1,04827 \cdot 10^{-2}$	$1,10627 \cdot 10^{-5}$	$3,62316 \cdot 10^{-5}$	$2,24642 \cdot 10^{-5}$
p	$9,90714 \cdot 10^{-3}$	$1,16704 \cdot 10^{-2}$	$1,05011 \cdot 10^{-2}$	$8,72665 \cdot 10^{-5}$	$2,73943 \cdot 10^{-4}$	$1,83622 \cdot 10^{-5}$
q	$9,91727 \cdot 10^{-3}$	$1,16657 \cdot 10^{-2}$	$1,04824 \cdot 10^{-2}$	$2,21450 \cdot 10^{-6}$	$7,25250 \cdot 10^{-6}$	$4,49214 \cdot 10^{-6}$

Taula 3.4 Paràmetres simulacions taula 3.3

	Abast	Sostre	Residu	Model		Abast	Sostre	Residu	Model
j	1,6	10^{-7}	0	Expon.	n	3	10^{-8}	0	Esfèric
k	1,6	10^{-7}	10^{-10}	Esfèric	o	10	10^{-8}	0	Esfèric
l	1,6	10^{-8}	$1 \cdot 10^{-9}$	Esfèric	p	1,6	10^{-7}	0	Esfèric
m	1,6	10^{-8}	0	Esfèric	q	250	10^{-8}	0	Esfèric

3.3. Resultats

Encara que a les taules anteriors només hi ha una mostra de les simulacions realitzades, es van realitzar moltes més. Aquí només s'han inclòs els valors que donen resultats semblants als esperats i errors més petits. En pujar o baixar d'aquest rang de nombres tant per a l'abast com per al sostre o el residu, les dades obtingudes no són tant adients com les mostrades.

Un cop es va trobar el rang més adequat es van anar ajustant les dades fins trobar els paràmetres més adients i amb menys error.

3.3.1. Resultats 2D

Per a les simulacions en 2 dimensions s'han provat moltes combinacions possibles i finalment s'ha decidit que el model més adequat serà l'esfèric, ja que l'exponencial sempre té un error més alt i el gaussià treballa fora dels rangs desitjats per al *tip clearance*, cosa que fa impossible obtenir resultats.

Encara que en variar el sostre i el residu, el resultat no varia de manera apreciable, perquè varia en ordre de 10^{-10} , l'error sí que canvia de manera visible perquè es treballa amb valors molt més petits. L'abast en canvi, sí que fa variar els resultats significativament, tant com per poder observar-se en la gràfica.

Finalment les millors simulacions s'han realitzat per un abast d'1,8, un sostre de 10^{-10} , un residu nul, ja que en la tendència dels valors es pot veure que és 0 per a una de distància 0 i, com ja s'ha dit, el model esfèric.

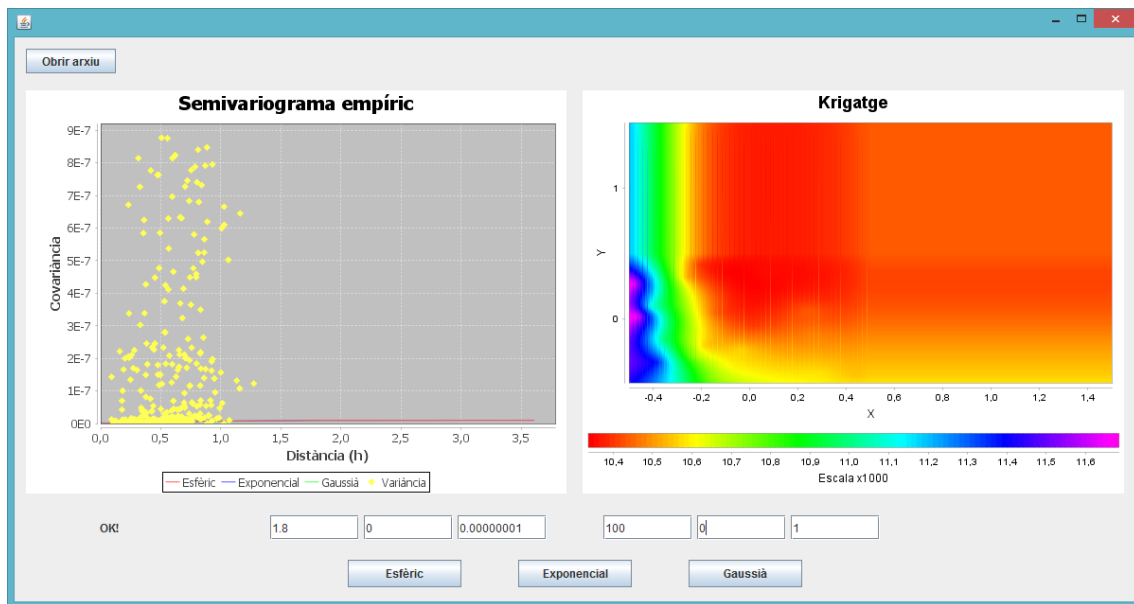


Fig 3.6 Model més ajustat per a 2D

En canvi els valors més petits dels resultats han estat amb un abast de 0,7 i un sostre de 10^{-8} . On per a tots els punts amb aquestes dades ha donat un valor mínim.

Es pot observar a la **Fig 3.7** que el model seleccionat és un model amb una pendent moderada, per al sostre seleccionat, l'abast és molt gran, perquè encara que les dades conegudes que són més a prop del punt que es vol calcular tenen més pes que les llunyanes, aquestes també influeixen.

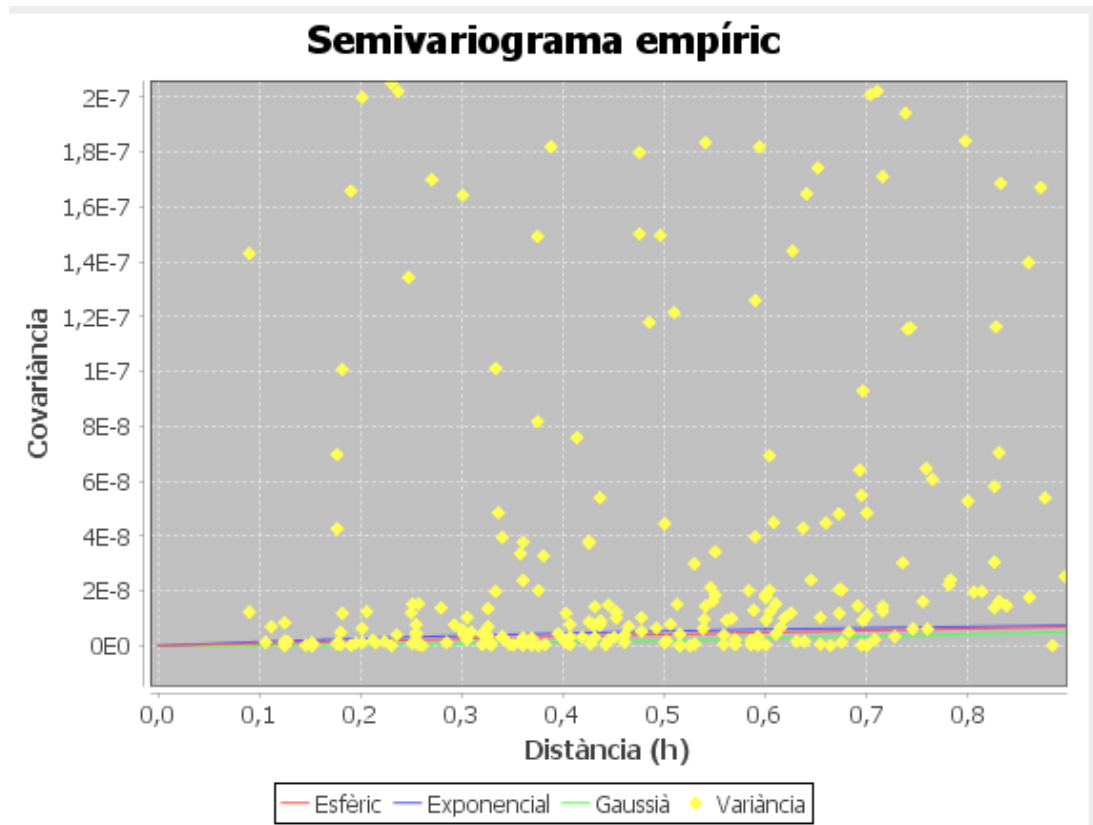


Fig 3.7 Model de semivariograma seleccionat

3.3.2. Resultats 5D

En els resultats calculats amb les dades de cinc paràmetres es pot observar que el *tip clearance* és més petit que per a les dades 2D, perquè el càlcul és capaç d'ajustar més el resultat, ja que té més paràmetres que hi influeixen, per tant la precisió serà més alta. És evident que encara que la exactitud escollida no és tant alta com per a les dades de dos paràmetres, ja que per aquella s'havia escollit 100 d'exactitud i per aquestes només 5, el nombre de dades d'entrada i de sortida és molt més alt, ja que per a cada punt hi ha 5 paràmetres.

Després de comprovar les diverses simulacions realitzades i anar ajustant els rangs i els paràmetres, es troba que el millor abast és de 250, amb un sostre molt petit de 10^{-8} , sense residu i amb el model esfèric, ja que novament el model gaussià no permet realitzar càlculs amb aquestes dades i l'exponencial no és tant precís.

Es pot observar que amb un residu més gran l'error era més petit, però no tant petit com per a un model sense residu. Això és perquè requereix un model on la pendent inicial no sigui molt pronunciada, és a dir, que les dades més properes no tenen una influència massa més gran que les més allunyades sinó que és una mica més homogeni el pes.

CONCLUSIONS

En començar el treball es pretenien assolir diversos objectius, entre ells, comprendre el funcionament de la interpolació amb el mètode de kriging i aplicarla a un cas amb dades reals i és el que ha quedat reflexat al llarg de tot el document.

Després d'assolir els coneixements necessaris sobre aquest mètode, que era una part important dels objectius, s'ha aconseguit realitzar un programa sencer, després d'aproximadament 180 hores de programació, que realitza els càlculs de manera correcta i és capaç de gestionar una gran quantitat de dades a la vegada per poder realitzar aproximacions bastant precises, amb errors de l'ordre de milionèsimes de la unitat utilitzada a les dades.

Per a aconseguir aquest altre objectiu s'han hagut d'aprendre molts aspectes desconeguts anteriorment del llenguatge, com per exemple fer servir Netbeans per a mostrar la part visual, ja que es va veure que era el més intuïtiu, aprendre el funcionament de JFreeChart que és la llibreria utilitzada per a incloure les gràfiques, aprendre a importar i exportar arxius, utilitzar un gràfic de colors amb una escala i en general controlar la unió entre el codi i la part gràfica, és a dir que el codi sigui apte per a poder mostrar la part visual i no només que es realitzin els càlculs a nivell intern del programa.

També s'han solucionat diversos problemes al llarg del procés de programació, com per exemple el càlcul de la inversa de les matrius que si es realitzava amb el mètode de Gauss-Jordan el temps s'incrementava de manera exponencial i era impossible tractar amb més de 7 punts a l'hora, finalment es va optar per utilitzar una llibreria que realitza el càlcul de manera òptima.

Un altre problema que es va haver de solucionar va ser en el moment de generar els punts finals, ja que com que la quantitat de coordenades és variable i depèn del fitxer d'entrada, es va haver de realitzar una funció recursiva que realitza el càlcul tantes vegades com coordenades hi ha.

Encara que el gràfic de colors de la dreta per a les dades d'aquest projecte no té massa importància, es va creure convenient incloure'l perquè en altres tipus de càlculs és molt interessant veure els resultats de manera gràfica, ja que permet una interpretació ràpida sense la necessitat d'analitzar les dades numèriques.

Aquest projecte és la base per a poder realitzar un projecte molt més ampliat amb col·laboració amb CIMNE, on es podrien analitzar una gran quantitat de paràmetres, ja que les dades d'entrada es poden introduir amb un número il·limitat de dimensions, encara que tenint en compte l'augment de temps a l'hora de realitzar el càlcul.

Per això el programa està realitzat en Java, per poder-lo modificar, ja que a l'hora de que algú hagi d'ampliar-lo o fer-li ajusts serà molt més senzill per l'estructura que segueix aquest tipus de llenguatge.

En un futur es podria implementar una millora que permetés no agafar totes les dades alhora, sinó que es poguessin agafar per trams, és un recurs molt utilitzat en les interpolacions i al qual es va optar al començament, per els problemes esmentats amb les inverses de les matrius. Aquesta millora optimitzaria molt el temps de càlcul però sacrificant la precisió. S'hauria de trobar un equilibri que dependria de la natura de les dades a tractar. En aquest cas no s'ha inclòs perquè no s'ha vist necessari, ja que no s'utilitzen més de 5 dimensions.

.

BIBLIOGRAFIA

- [1] W. Z. Michalak, *GEO 641* [en línia]. *Spatial Analyst II* Toronto: Ryerson University, 2004 [Consulta: 17 agost 2015] Disponible a: <<http://www.geography.ryerson.ca/>>
- [2] ArcGIS Resource Center, *Cómo funciona Kriging* [en línia]. *Desktop 10* Madrid: Esri, 2012 [Consulta: 21 maig 2015] Disponible a: <<http://help.arcgis.com/es/arcgisdesktop/10.0/>>
- [3] D. G. Krige, *A statistical approach to some mine valuations and allied problems at the Witwatersrand* Master's thesis of the University of Witwatersrand, South Africa, (1951)
- [4] N. Cressie, "The origins of kriging," *Mathematical Geology*, vol. 22, pp. 239-252, (1990)
- [5] G. Matheron, "Kriging, or Polynomial interpolation procedures? A contribution to polemics in mathematical geology," *Ecole Nationale Supérieure des Mines*, vol. LXX, pp. 204-244. (1967)
- [6] A. Ratle, *Optimal sampling strategies for learning a fitness model* [en línia] Québec: Département de génie mécanique, Université de Sherbrooke, 1999 [Consulta: 27 d'agost 2015] Disponible a: <<http://ieeexplore.ieee.org/>>
- [7] T. Hengl, *Kriging algorithm* [en línia] *Spatial analyst* Amsterdam: ILWIS, 2014. [Consulta: 17 maig 2015] Disponible a: < http://spatial-analyst.net/ILWIS/htm/ilwisapp/kriging_algorithm.htm>
- [8] T. M. M. J. J. K. i. F. M. T. W. Simpson, "Kriging models for global approximation in simulation-based multidisciplinary design optimization," *AIAA Journal*, vol. 39, núm. 12, (2001)
- [9] Dr. J. Nunes, *Semivariograma* [en línia] Barcelona: Institut Cartogràfic i Geològic de Catalunya, 2013. [Consulta: 16 agost 2015] Disponible a: <<http://www.icc.cat/>>
- [10] M. B. i M. Backes, *Variogram or semivariogram? Understanding the variances in a variogram* [en línia] Nova York: Springer Science+Business Media, 2008. [Consulta: 16 agost 2015] Disponible a: <ink.springer.com/article/10.1007>
- [11] G. Bohling, *Introduction to geostatistics and variogram analysis*, [en línia] Kansas: C&PE 940, 2005. [Consulta: 16 agost 2015] Disponible a: <people.ku.edu/~4gbohling/cpe940/Variograms.pdf>

- [12] V. Gandhi, *Semivariogram modeling*, [en línia] Minneapolis: Department of Computer Science and Engineering, University of Minnesota, 2008. [Consulta: 16 agost 2015] Disponible a: <www-users.cs.umn.edu>
- [13] S. N. L. N. Cressie, «The asymptotic distribution of REML Estimators,» *Journal of multivariate Analysis*, vol. 45, núm. 2, pp. 217-233, (1993)
- [14] J. M. Marín, *Introducción a las redes neuronales aplicadas* [en línia] Madrid: Universidad Carlos III, 2011. [Consulta: 9 setembre 2015] Disponible a : <halweb.uc3m.es/esp/Personal/personas/jmmarin>
- [15] V. Yepes, *Metodología de superficies de respuesta, gran alternativa para incrementar la productividad de sus procesos* [en línea] Cali: Centro de ingeniería de la calidad, 2013. [Consulta: 9 setembre 2015] Disponible a: <www.cicalidad.com>
- [16] Wikipedia, *Interpolation* [en línia] Wikipedia the free encyclopedia, 2015. [Consulta: 3 setembre 2015] Disponible a: <<https://en.wikipedia.org/wiki/interpolation>>
- [17] J. Pons,[power point] *Surrogate model: Data Analysis*, CIMNE: Barcelona, (2015)
- [18] A. S. Foundation, *Commons Math* [en línia] *The Apache Commons Mathematics Library* Salt Lake City: ASF, 2015 [Consulta: 23 agost 2015] Disponible a: <commons.apache.org/proper/commons-math/>
- [19] P. M. Barbosa, *Introdução aos métodos de estimação espacial para confecção de mapas* [en línea] Rio Claro; Unesp, 2000 [Consulta: 24 agost 2015] Disponible a: <www.rc.unesp.br/igce/aplicada/>
- [20] N. Cressie, *Statistics for spatial data* [en línea] Ames: Iowa State University, Wiley-Interscience, 1993. [Consulta: 24 agost 2015] Disponible a: <<https://rongxie.files.wordpress.com>>



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castellet

UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANNEXOS

TÍTOL DEL TFC: Models substitutius per a modelat de dades aeronàutiques

TITULACIÓ: Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació

AUTOR: Alicia Herreros Suárez

DIRECTOR: Jordi Pons i Prats

SUPERVISOR: Francisco Javier Mora Serrano

DATA: 12 de setembre de 2015

ANNEX 1: CODI EN JAVA

Main.java

```
package kriging;

import java.io.IOException;

public class Main {
    public static void main(String[] args) throws IOException {
        Kriging kriging = new Kriging();
        kriging.kriging();
    }
}
```

Kriging.java

```
package kriging;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.LinkedList;
import org.apache.commons.math3.linear.LUDecomposition;
import org.apache.commons.math3.linear.MatrixUtils;
import org.apache.commons.math3.linear.RealMatrix;
import org.jfree.data.xy.DefaultXYZDataset;
import org.jfree.data.xy.XYDataset;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
import org.jfree.data.xy.XYZDataset;

public class Kriging {

    private BufferedWriter bw;
    private final KrigingFrame kf = new KrigingFrame();
    private LinkedList<Punto> in, cor, mincor, res, err;
    private LinkedList<Distancia> dis;
    private Parametros param = new Parametros();
    private double c0, c1, a, begin, end, accuracy, cnt;
    private double Cd[][], D[], weight[], result[], maximo=0,
minimo=999999999;
    private double V[][];
    private int multiplicar, modo;

    public void kriging() throws IOException {
        while(true){
            kf.setVisible(true);
            //Primera gr fica
            if(kf.getArchivo()){
                inicializacion();
                for(int i=0; i<in.size(); i++){
                    for(int j=i+1; j<in.size(); j++){
                        Cd[i][j] = distancia(in.get(i), in.get(j));
                        Distancia d = new Distancia
(in.get(i), in.get(j), Cd[i][j]);
                        dis.add(d);
                    }
                }
            }
        }
    }
}
```

```

    }
    ordenarDistancias(dis);
    coVarianza();
    kf.prueba(createDataset(false));
}
//Primer gràfica (2)
if(kf.getCampos1()){
    Parametros p = new Parametros (kf.getOpcion1());
    if(!p.comparar(param)){
        try {
            parametrizacion(kf.getOpcion1());
            kf.prueba2(createDataset(true));
        }catch(Exception e){System.out.println("Error");}
        param = new Parametros(p);
    }
}
//Segunda gràfica
if (kf.getBoton() && kf.getCampos2()){
    kf.setVisible(true);
    try {
        param = new Parametros();
        param=kf.getOpcion2();
    }catch(Exception e){}
    parametrizacion(param);
    modo = param.getModo();
    inicializacion2();
    varianza();
    RealMatrix m = MatrixUtils.createRealMatrix(V);
    try{
        RealMatrix pInverse = new
LUdecomposition(m).getSolver().getInverse();
        V=pInverse.getData();
    }catch(Exception e){
        kf.setTexto("Matriu singular");
    }
    for (int
contador=0;contador<mincor.size();contador++){
        kf.setTiempo((int) (contador*100/mincor.size()));
        krig_result(contador);
    }
    if(res.get(0).getCoordenadas().size()==3){
        kf.prueba3(createDataset2(), maximo, minimo,
multiplicar);
    }
    escribirResultado();
    escribirError();
    kf.setTiempo((int) 100);
}
}
}
//Primera Gràfica
public void inicializacion(){
    in = new LinkedList<>();
    in = kf.getInput();
    Cd = new double [in.size()+1][in.size()+1];
    dis = new LinkedList<>();
    cor = new LinkedList<>();
}
public double distancia(Punto a, Punto b){
    double d=0;
    for(int k=0;k<a.getCoordenadas().size()-1;k++){

```

```

        d = d + Math.pow(a.getCoordenada(k).getVal() -
b.getCoordenada(k).getVal(),2);
    }
    return Math.sqrt(d);
}
}
public void ordenarDistancias(LinkedList<Distancia> distancia){
    Distancia aux;
    for (int i=0;i<distancia.size()-1;i++){
        for (int j=0;j<distancia.size()-1;j++){
            if (distancia.get(j).getDis() >
distancia.get(j+1).getDis()){
                aux = distancia.get(j+1);
                distancia.set(j+1, distancia.get(j));
                distancia.set(j, aux);
            }
        }
    }
}
public void coVarianza(){
    double media;
    int k=1, i, j;
    for (i=0;i<dis.size();i++){
        media =
Math.pow(dis.get(i).getA().getCoordenadas().getLast().getVal()-
dis.get(i).getB().getCoordenadas().getLast().getVal(),2);
        for (j=i+1;j<dis.size();j++){
            if (dis.get(i).getDis()==dis.get(j).getDis()){
                k++;
                media = media +
Math.pow(dis.get(i).getA().getCoordenadas().getLast().getVal() -
dis.get(i).getB().getCoordenadas().getLast().getVal(),2);
            }else{
                break;
            }
        }
        media = (media/k) / 2;
        Punto p= new Punto();
        Coordenada x = new Coordenada(dis.get(i).getDis());
        Coordenada y = new Coordenada(media);
        p.addCoordenada(x);
        p.addCoordenada(y);
        cor.add(p);
        if (maximo < dis.get(i).getDis()){
            maximo = dis.get(i).getDis();
        }
        if (minimo > dis.get(i).getDis()){
            minimo = dis.get(i).getDis();
        }
        i=j-1;
        k=1;
    }
}
}
public XYDataset createDataset(boolean b) {
    XYSeriesCollection resultado = new XYSeriesCollection();
    XYSeries series = new XYSeries("Variància");
    for (int i = 0; i < cor.size(); i++) {
        double x = cor.get(i).getCoordenada(0).getVal();
        double y = cor.get(i).getCoordenada(1).getVal();
        series.add(x, y);
    }
}

```

```

    }
    if(b){
        double x,y;
        //Spherical
        XYSeries series2 = new XYSeries("Esfèric");
        x = a*2;
        if (x <= a){
            y = c0 + c1 *(1.5*x/a - 0.5*(x/a)*(x/a)*(x/a));
        }else{
            y = c0 + c1;
        }
        series2.add(x, y);
        for (int i = 1; i < 20; i++) {
            x = a/i;
            if (x <= a){
                y = c0 + c1 *(1.5*x/a - 0.5*(x/a)*(x/a)*(x/a));
            }else{
                y = c0 + c1;
            }
            series2.add(x, y);
        }
        x = 0;
        if (x <= a){
            y = c0 + c1 *(1.5*x/a - 0.5*(x/a)*(x/a)*(x/a));
        }else{
            y = c0 + c1;
        }
        series2.add(x, y);
        //Exponential
        XYSeries series3 = new XYSeries("Exponencial");
        x = a*2;
        y = c0 + c1 *(1 - Math.exp(-Math.E*x/a));
        series3.add(x, y);
        for (int i = 1; i < 20; i++) {
            x = a/i;
            y = c0 + c1 *(1 - Math.exp(-Math.E*x/a));
            series3.add(x, y);
        }
        x = 0;
        y = c0 + c1 *(1 - Math.exp(-Math.E*x/a));
        series3.add(x, y);
        //Gauss
        XYSeries series4 = new XYSeries("Gaussià");
        x = a*2;
        y = c0 + c1 *(1 - Math.exp(-Math.E*x*x/a/a));
        series4.add(x, y);
        for (int i = 1; i < 20; i++) {
            x = a/i;
            y = c0 + c1 *(1 - Math.exp(-Math.E*x*x/a/a));
            series4.add(x, y);
        }
        x = 0;
        y = c0 + c1 *(1 - Math.exp(-Math.E*x*x/a/a));
        series4.add(x, y);
        resultado.addSeries(series2);
        resultado.addSeries(series3);
        resultado.addSeries(series4);
    }
    resultado.addSeries(series);
    return resultado;
}

```



```

//Segunda Gráfica
public void parametrizacion(Parametros param) throws IOException{
    a = param.getA();
    c0 = param.getC0();
    c1 = param.getC1();
    maximo=0;
    minimo=999999999;
    multiplicar = 1;
    begin = param.getBegin();
    end = param.getEnd();
    accuracy = param.getAccuracy();
    mincor = new LinkedList<>();
}
public void inicializacion2() throws IOException{
    res = new LinkedList<>();
    err = new LinkedList<>();
    V = new double [in.size()+1][in.size()+1];
    D = new double [in.size()+1];
    weight = new double [in.size()+1];
    result = new double[2];
    multiplicar = 1;
    cnt = end/accuracy;
    for(int j=0;j<=accuracy;j++) {
        Punto p = new Punto();
        Coordenada c = new
Coordenada(Math rint((cnt*j+begin)*100000)/100000);
        p.addCoordenada(c);
        if(in.get(0).getCoordenadas().size()>2){
            puntosFinales(p,1,in.get(0).getCoordenadas().size()-
2);
        }else{
            mincor.add(p);
        }
    }
}
public void puntosFinales(Punto punto, int dim, int dimensiones){
    for (int j=1;j<=accuracy;j++) {
        Coordenada c = new
Coordenada(Math rint((cnt*j+begin)*100000)/100000);
        punto.addCoordenada(c);
        for (int i=dim; i<dimensiones;i++){
            puntosFinales(punto, dim+1,dimensiones);
        }
        if(dim==dimensiones){
            Punto p2 = new Punto();
            for(int k=0;k<punto.getCoordenadas().size();k++){
                p2.addCoordenada(punto.getCoordenada(k));
            }
            mincor.add(p2);
        }
        punto.getCoordenadas().removeLast();
    }
}
public void varianza(){
    int i,j;
    /*Cálculo de las covarianzas entre las muestras*/
    for (i=0; i<in.size(); i++){
        V[i][in.size()] = 1;
        V[in.size()][i] = 1;
        for (j=i; j<in.size(); j++){
            switch (modo){

```

```

        case 1: /*Esferico*/
            if (Cd[i][j] <= a){
                V[i][j] = V[j][i] = c0 + c1
                *(1.5*Cd[i][j]/a - 0.5*(Cd[i][j]/a)*(Cd[i][j]/a)*(Cd[i][j]/a));
            }else{
                V[i][j] = V[j][i] = c0 + c1;
            }
            break;
        case 2: /*Exponencial*/
            V[i][j] = V[j][i] = c0 + c1 *(1 - Math.exp(-
Math.E*Cd[i][j]/a));
            break;
        case 3: /*Gauss*/
            V[i][j] = V[j][i] = c0 + c1 *(1 - Math.exp(-
Math.E*Cd[i][j]*Cd[i][j]/a/a));
            break;
    }

    }

    V[in.size()][i] = 0;
}

public void krig_result(int contador){
    double h;
    int i,j;
    /*Calculo de la covarianza entre la ley del bloque y cada una
de las muestras*/
    for (i=0; i < in.size() ; i++) {
        h = distancia(in.get(i),mincor.get(contador));
        switch( modo ) {
            case 1 : /* Esferico*/
                if ( h < a ){
                    D[i] = c0 + c1 * (1.5*h/a -
0.5*Math.pow(h/a,3));
                }
                else
                    D[i] = c0 + c1;
                break;
            case 2 : /* Exponencial*/
                D[i] = c0 + c1 * (1 - Math.exp(-Math.E*h/a));
                break;
            case 3 : /* Gauss*/
                D[i] = c0 + c1 * (1 - Math.exp(-Math.E*h*h/a/a));
                break;
        }
    }
    D[i] = 1;
    result[0] = 0;
    /* result[0] : kriging result */
    /* result[1] : error variance result */
    for ( i = 0 ; i < in.size()+1; i++) {
        weight[i] = 0;
        for ( j = 0 ; j<in.size()+1; j++){
            weight[i] += V[i][j] * D[j];
        }
        if(i<in.size()){
            result[0] += weight[i] *
in.get(i).getCoordenadas().getLast().getVal();
        }
    }
    Punto p = new Punto();

```

```

        for(int k=0;k<mincor.get(0).getCoordenadas().size();k++){
            Coordenada c = new
Coordenada(mincor.get(contador).getCoordenada(k).getVal());
            p.addCoordenada(c);
        }
        Coordenada c = new Coordenada(result[0]);
        p.addCoordenada(c);
        res.add(p);
        result[0] = ( result[0] >= 0 ) ? result[0] : 0;
        result[1] = 0;
        for ( i = 0 ;i < in.size()+1;i++) {
            result[1] += weight[i] * D[i];
        }
        result[1] += weight[in.size()];
        result[1] = (float) Math.sqrt(result[1]);
        Punto p2 = new Punto();
        for(int e=0;e<p.getCoordenadas().size()-1;e++){
            p2.addCoordenada(p.getCoordenada(e));
        }
        Coordenada c2 = new Coordenada(result[1]);
        p2.addCoordenada(c2);
        err.add(p2);
    }
    private XYZDataset createDataset2() throws IOException {
        DefaultXYZDataset resultado = new DefaultXYZDataset();
        double[] xdata = new double[res.size()];
        double[] ydata = new double[res.size()];
        double[] zdata = new double[res.size()];
        double[][] data = new double[][] {xdata,ydata,zdata};
        boolean b = false;
        double min = 9999999;
        for (int i = 0; i < res.size(); i++) {

if(Math.abs(res.get(i).getCoordenadas().getLast().getVal()) <1 &&
            res.get(i).getCoordenadas().getLast().getVal()
!=0){
                b = true;
                if
(min>Math.abs(res.get(i).getCoordenadas().getLast().getVal())
                    &&
res.get(i).getCoordenadas().getLast().getVal()!=0){
                    min=
Math.abs(res.get(i).getCoordenadas().getLast().getVal());
                }
            }
        }
        if(b){
            while(min<10){
                multiplicar = multiplicar *10;
                min = min * 10;
            }
        }
        for(int i = 0;i<res.size();i++){
            if (maximo <
(res.get(i).getCoordenadas().getLast().getVal()*multiplicar){
                maximo =
res.get(i).getCoordenadas().getLast().getVal()*multiplicar;
            }
            if (minimo >
(res.get(i).getCoordenadas().getLast().getVal()*multiplicar)){

```

```

        minimo =
res.get(i).getCoordenadas().getLast().getVal()*multiplicar;
    }
    data[0][i] = res.get(i).getCoordenada(0).getVal();
    data[1][i] = res.get(i).getCoordenada(1).getVal();
    data[2][i] =
res.get(i).getCoordenada(2).getVal()*multiplicar;
    }
    resultado.addSeries("Resultado",data);
    return resultado;
}
public void escribirResultado() throws IOException{
    bw = new BufferedWriter(new FileWriter("result.txt",false));
    for (int i = 0; i < res.size(); i++) {
        for (int j = 0; j<
res.get(0).getCoordenadas().size();j++){
            bw.write("x"+j+" =
"+res.get(i).getCoordenada(j).getVal()+"\t");
        }
        bw.newLine();
    }
    bw.close();
}
public void escribirError() throws IOException{
    bw = new BufferedWriter(new FileWriter("error.txt",false));
    for (int i = 0; i < err.size(); i++) {
        for (int j = 0; j<
err.get(0).getCoordenadas().size();j++){
            bw.write("x"+j+" =
"+err.get(i).getCoordenada(j).getVal()+"\t");
        }
        bw.newLine();
    }
    bw.close();
}
}
}

```

Coordenada.java

```

package kriging;

public class Coordenada {
    private double valor;

    public Coordenada(){
    }
    public Coordenada(double i){
        valor = i;
    }

    public double getVal(){ return valor; }

    public void setVal(double i){ valor = i; }
}

```

Punto.java

```
package kriging;

import java.util.LinkedList;

public class Punto {
    private LinkedList<Coordenada> coordenadas = new LinkedList<>();

    public Punto(){

    }

    public LinkedList<Coordenada> getCoordenadas() { return
coordenadas; }
    public Coordenada getCoordenada(int i) { return
coordenadas.get(i); }

    public void setCoordenadas(LinkedList<Coordenada> i){
coordenadas = i; }
    public void addCoordenada(Coordenada c) {
coordenadas.add(c); }

    @Override
    public String toString(){
        String s="";
        for (int i=0;i<coordenadas.size();i++){
            s = s+"c"+i+" :"+coordenadas.get(i).getVal()+" ";
        }
        return s;
    }
}
```

Parametros.java

```
package kriging;

public class Parametros {

    private int modo;
    private float a;
    private float c0;
    private float c1;
    private int accuracy;
    private int begin;
    private int end;

    public Parametros(){

    }
    public Parametros(int i, float j, float k, float l, int m, int n,
int o){
        modo=i;
        a=j;
        c0=k;
        c1=l;
        accuracy=m;
        begin=n;
    }
}
```

```

        end=o;
    }
    public Parametros(Parametros p){
        modo=p.getModo();
        a=p.getA();
        c0=p.getC0();
        c1=p.getC1();
        accuracy=p.getAccuracy();
        begin=p.getBegin();
        end=p.getEnd();
    }

    public int getModo(){           return modo;           }
    public float getA(){            return a;            }
    public float getC0(){           return c0;           }
    public float getC1(){           return c1;           }
    public int getAccuracy(){        return accuracy;     }
    public int getBegin(){           return begin;        }
    public int getEnd(){            return end;            }

    public void setModo(int i){      modo = i;           }
    public void setA(float i){       a = i;                }
    public void setC0(float i){      c0 = i;                }
    public void setC1(float i){      c1 = i;                }
    public void setAccuracy(int i){  accuracy = i;         }
    public void setBegin(int i){     begin = i;             }
    public void setEnd(int i){       end = i;               }

    public boolean comparar(Parametros a){
        if (a.getA()==this.a &&
            a.getC0()==this.c0 &&
            a.getC1()==this.c1 &&
            a.getAccuracy()==this.accuracy &&
            a.getBegin()==this.begin &&
            a.getEnd()==this.end){
            return true;
        }else{
            return false;
        }
    }
}

```

Distancia.java

```

package kriging;
public class Distancia {
    private Punto a;
    private Punto b;
    private double dis;

    public Distancia(){

    }
    public Distancia(Punto i, Punto j, double k){
        a=i;
        b=j;
        dis=k;
    }
}

```

```

    public Punto getA(){          return a;    }
    public Punto getB(){          return b;    }
    public double getDis(){       return dis; }

    public void setA(Punto i){    a=i;        }
    public void setB(Punto i){    b=i;        }
    public void setDis(double i){ dis=i;      }
}

```

KrigingFrame.java

```

package kriging;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.Paint;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.LinkedList;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import static jdk.nashorn.internal.runtime.JSType.isNumber;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.XYPlot;
import org.jfree.chart.renderer.LookupPaintScale;
import org.jfree.chart.renderer.xy.XYBlockRenderer;
import org.jfree.chart.renderer.xy.XYLineAndShapeRenderer;
import org.jfree.chart.title.PaintScaleLegend;
import org.jfree.data.xy.XYDataset;
import org.jfree.data.xy.XYZDataset;
import org.jfree.ui.RectangleEdge;
import org.jfree.ui.RectangleInsets;

public class KrigingFrame extends javax.swing.JFrame {

    public KrigingFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize
     the form.
     * WARNING: Do NOT modify this code. The content of this method is
     always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">
    private void initComponents() {

```

```

jPanel11 = new javax.swing.JPanel();
jButton2 = new javax.swing.JButton();
jPanel12 = new javax.swing.JPanel();
jPanel14 = new javax.swing.JPanel();
jButton1 = new javax.swing.JButton();
jPanel18 = new javax.swing.JPanel();
jButton3 = new javax.swing.JButton();
jPanel19 = new javax.swing.JPanel();
jButton4 = new javax.swing.JButton();
jPanel15 = new javax.swing.JPanel();
jPanel110 = new javax.swing.JPanel();
jPanel118 = new javax.swing.JPanel();
jLabel3 = new javax.swing.JLabel();
jPanel113 = new javax.swing.JPanel();
jTextField3 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jTextField11 = new javax.swing.JTextField();
jPanel111 = new javax.swing.JPanel();
jTextField8 = new javax.swing.JTextField();
jTextField9 = new javax.swing.JTextField();
jTextField10 = new javax.swing.JTextField();
jPanel114 = new javax.swing.JPanel();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMaximumSize(new java.awt.Dimension(12000, 6000));
setMinimumSize(new java.awt.Dimension(1200, 635));
setPreferredSize(new java.awt.Dimension(1200, 600));

jPanel11.setMaximumSize(new java.awt.Dimension(820, 50));
jPanel11.setMinimumSize(new java.awt.Dimension(820, 50));
jPanel11.setPreferredSize(new java.awt.Dimension(820, 50));
jPanel11.setLayout(new javax.swing.BoxLayout(jPanel11,
javax.swing.BoxLayout.LINE_AXIS));

jButton2.setText("Obrir arxiu");
jButton2.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jButton2ActionPerformed(evt);
    }
});
jPanel11.add(jButton2);

jPanel12.setBackground(new java.awt.Color(204, 204, 204));
jPanel12.setMaximumSize(new java.awt.Dimension(5680, 4000));
jPanel12.setMinimumSize(new java.awt.Dimension(569, 418));
jPanel12.setPreferredSize(new java.awt.Dimension(569, 418));
jPanel12.setLayout(new java.awt.BorderLayout());

jPanel14.setMaximumSize(new java.awt.Dimension(11550, 400));
jPanel14.setMinimumSize(new java.awt.Dimension(1155, 40));
jPanel14.setPreferredSize(new java.awt.Dimension(1155, 40));

jButton1.setText("Esfèric");
jButton1.setMaximumSize(new java.awt.Dimension(120, 29));
jButton1.setMinimumSize(new java.awt.Dimension(120, 29));
jButton1.setPreferredSize(new java.awt.Dimension(120, 29));

```



```

        jButton1.addActionListener(new java.awt.event.ActionListener()
        {
            public void actionPerformed(java.awt.event.ActionEvent
            evt) {
                jButton1ActionPerformed(evt);
            }
        });
        jPanel4.add(jButton1);

        jPanel8.setMaximumSize(new java.awt.Dimension(50, 25));
        jPanel8.setMinimumSize(new java.awt.Dimension(50, 25));
        jPanel8.setPreferredSize(new java.awt.Dimension(50, 25));
        jPanel4.add(jPanel8);

        jButton3.setText("Exponencial");
        jButton3.setMaximumSize(new java.awt.Dimension(120, 29));
        jButton3.setMinimumSize(new java.awt.Dimension(120, 29));
        jButton3.setPreferredSize(new java.awt.Dimension(120, 29));
        jButton3.addActionListener(new java.awt.event.ActionListener()
        {
            public void actionPerformed(java.awt.event.ActionEvent
            evt) {
                jButton3ActionPerformed(evt);
            }
        });
        jPanel4.add(jButton3);

        jPanel9.setMaximumSize(new java.awt.Dimension(50, 25));
        jPanel9.setMinimumSize(new java.awt.Dimension(50, 25));
        jPanel9.setPreferredSize(new java.awt.Dimension(50, 25));
        jPanel4.add(jPanel9);

        jButton4.setText("Gaussià");
        jButton4.setMaximumSize(new java.awt.Dimension(120, 29));
        jButton4.setMinimumSize(new java.awt.Dimension(120, 29));
        jButton4.setPreferredSize(new java.awt.Dimension(120, 29));
        jButton4.addActionListener(new java.awt.event.ActionListener()
        {
            public void actionPerformed(java.awt.event.ActionEvent
            evt) {
                jButton4ActionPerformed(evt);
            }
        });
        jPanel4.add(jButton4);

        jPanel5.setBackground(new java.awt.Color(204, 204, 204));
        jPanel5.setMaximumSize(new java.awt.Dimension(5690, 4000));
        jPanel5.setMinimumSize(new java.awt.Dimension(569, 418));
        jPanel5.setPreferredSize(new java.awt.Dimension(569, 418));
        jPanel5.setLayout(new java.awt.BorderLayout());

        jPanel10.setMaximumSize(new java.awt.Dimension(820, 40));
        jPanel10.setMinimumSize(new java.awt.Dimension(820, 40));
        jPanel10.setPreferredSize(new java.awt.Dimension(820, 40));

        jPanel18.setMaximumSize(new java.awt.Dimension(50, 25));
        jPanel18.setMinimumSize(new java.awt.Dimension(50, 25));
        jPanel18.setPreferredSize(new java.awt.Dimension(50, 25));
        jPanel10.add(jPanel18);

        jLabel3.setMaximumSize(new java.awt.Dimension(120, 20));

```

```

jLabel13.setMinimumSize(new java.awt.Dimension(120, 20));
jLabel13.setPreferredSize(new java.awt.Dimension(120, 20));
jPanel10.add(jLabel13);

jPanel13.setMaximumSize(new java.awt.Dimension(50, 25));
jPanel13.setMinimumSize(new java.awt.Dimension(50, 25));
jPanel13.setPreferredSize(new java.awt.Dimension(50, 25));
jPanel10.add(jPanel13);

jTextField3.setText("Abast (A)");
jTextField3.setMaximumSize(new java.awt.Dimension(94, 26));
jTextField3.setMinimumSize(new java.awt.Dimension(94, 26));
jTextField3.setPreferredSize(new java.awt.Dimension(94, 26));
jTextField3.addFocusListener(new java.awt.event.FocusAdapter()
{
    public void focusGained(java.awt.event.FocusEvent evt) {
        jTextField3FocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        jTextField3FocusLost(evt);
    }
});
jPanel10.add(jTextField3);

jTextField7.setText("Residu (c0)");
jTextField7.setMaximumSize(new java.awt.Dimension(94, 26));
jTextField7.setMinimumSize(new java.awt.Dimension(94, 26));
jTextField7.setPreferredSize(new java.awt.Dimension(94, 26));
jTextField7.addFocusListener(new java.awt.event.FocusAdapter()
{
    public void focusGained(java.awt.event.FocusEvent evt) {
        jTextField7FocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        jTextField7FocusLost(evt);
    }
});
jPanel10.add(jTextField7);

jTextField11.setText("Sostre (c1)");
jTextField11.setMaximumSize(new java.awt.Dimension(94, 26));
jTextField11.setMinimumSize(new java.awt.Dimension(94, 26));
jTextField11.setPreferredSize(new java.awt.Dimension(94, 26));
jTextField11.addFocusListener(new
java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        jTextField11FocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        jTextField11FocusLost(evt);
    }
});
jPanel10.add(jTextField11);

jPanel11.setMaximumSize(new java.awt.Dimension(50, 25));
jPanel11.setMinimumSize(new java.awt.Dimension(50, 25));
jPanel11.setPreferredSize(new java.awt.Dimension(50, 25));
jPanel10.add(jPanel11);

jTextField8.setText("Exactitud");
jTextField8.setMaximumSize(new java.awt.Dimension(94, 26));

```

```

jTextField8.setMinimumSize(new java.awt.Dimension(94, 26));
jTextField8.setPreferredSize(new java.awt.Dimension(94, 26));
jTextField8.addFocusListener(new java.awt.event.FocusAdapter()
{
    public void focusGained(java.awt.event.FocusEvent evt) {
        jTextField8FocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        jTextField8FocusLost(evt);
    }
});
jPanell10.add(jTextField8);

jTextField9.setText("Inici");
jTextField9.setMaximumSize(new java.awt.Dimension(94, 26));
jTextField9.setMinimumSize(new java.awt.Dimension(94, 26));
jTextField9.setPreferredSize(new java.awt.Dimension(94, 26));
jTextField9.addFocusListener(new java.awt.event.FocusAdapter()
{
    public void focusGained(java.awt.event.FocusEvent evt) {
        jTextField9FocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        jTextField9FocusLost(evt);
    }
});
jPanell10.add(jTextField9);

jTextField10.setText("Final");
jTextField10.setMaximumSize(new java.awt.Dimension(94, 26));
jTextField10.setMinimumSize(new java.awt.Dimension(94, 26));
jTextField10.setPreferredSize(new java.awt.Dimension(94, 26));
jTextField10.addFocusListener(new
java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        jTextField10FocusGained(evt);
    }
    public void focusLost(java.awt.event.FocusEvent evt) {
        jTextField10FocusLost(evt);
    }
});
jTextField10.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent
evt) {
        jTextField10ActionPerformed(evt);
    }
});
jPanell10.add(jTextField10);

jPanell14.setMaximumSize(new java.awt.Dimension(50, 25));
jPanell14.setMinimumSize(new java.awt.Dimension(50, 25));
jPanell14.setPreferredSize(new java.awt.Dimension(50, 25));
jPanell10.add(jPanell14);

jLabell1.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);

jLabell1.setHorizontalTextPosition(javax.swing.SwingConstants.RIGHT);
jLabell1.setMaximumSize(new java.awt.Dimension(120, 20));
jLabell1.setMinimumSize(new java.awt.Dimension(120, 20));

```

```

jLabel1.setPreferredSize(new java.awt.Dimension(120, 20));
jPanel10.add(jLabel1);
jLabel1.getAccessibleContext().setAccessibleName("JLabel1");

jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
jLabel2.setMaximumSize(new java.awt.Dimension(50, 20));
jLabel2.setMinimumSize(new java.awt.Dimension(50, 20));
jLabel2.setPreferredSize(new java.awt.Dimension(50, 20));
jPanel10.add(jLabel2);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                layout.createSequentialGroup()

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
                        .TRAILING)
                        .addComponent(jPanel4,
                            javax.swing.GroupLayout.Alignment.LEADING,
                            javax.swing.GroupLayout.DEFAULT_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jPanel10,
                            javax.swing.GroupLayout.Alignment.LEADING,
                            javax.swing.GroupLayout.DEFAULT_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
                            layout.createSequentialGroup()
                                .addComponent(jPanel12,
                                    javax.swing.GroupLayout.DEFAULT_SIZE,
                                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addGap(17, 17, 17)
                                .addComponent(jPanel5,
                                    javax.swing.GroupLayout.DEFAULT_SIZE,
                                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                            .addGap(17, 17, 17)
                        .addComponent(jPanel11,
                            javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE,
                            javax.swing.GroupLayout.PREFERRED_SIZE)

                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jPanel5,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addGap(18, 18, 18)
    .addComponent(jPanel10,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(7, 7, 7)
    .addComponent(jPanel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(9, 9, 9))
);

pack();
} // </editor-fold> // GEN-END: initComponents

private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) { // GEN-FIRST: event_jButton2ActionPerformed
String cadena;
String[] partes;
try
{
    puntos.clear();
    /**llamamos el metodo que permite cargar la ventana*/
    JFileChooser file=new JFileChooser();
    file.showOpenDialog(this);
    /**abrimos el archivo seleccionado*/
    File abre=file.getSelectedFile();
    if(abre!=null)
    {
        FileReader archivos=new FileReader(abre);
        BufferedReader lee=new BufferedReader(archivos);
        while((cadena=lee.readLine())!=null)
        {
            partes = cadena.split(";");
            Punto p = new Punto();
            for(int j=0;j<partes.length;j++){
                Coordenada c = new
Coordenada(Float.parseFloat(partes[j]));
                p.addCoordenada(c);
            }
            puntos.add(p);
        }
        lee.close();
    }
}
catch(IOException ex)
{
    JOptionPane.showMessageDialog(null,ex+" " +
"\nNo se ha encontrado el archivo",
"ADVERTENCIA!!!",JOptionPane.WARNING_MESSAGE);
}
archivo = true;
}

```

```

    } //GEN-LAST:event_jButton2ActionPerformed

    private void jButton4ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jButton4ActionPerformed
        opcion = 3;
        boton = true;
    } //GEN-LAST:event_jButton4ActionPerformed

    private void jButton3ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jButton3ActionPerformed
        opcion = 2;
        boton = true;
    } //GEN-LAST:event_jButton3ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jButton1ActionPerformed
        opcion = 1;
        boton = true;
    } //GEN-LAST:event_jButton1ActionPerformed

    private void
jTextField10ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jTextField10ActionPerformed

    } //GEN-LAST:event_jTextField10ActionPerformed

    private void jTextField10FocusLost(java.awt.event.FocusEvent evt)
{ //GEN-FIRST:event_jTextField10FocusLost
        if ("".equals(jTextField10.getText())) {
            jTextField10.setText("Final");
        }
    } //GEN-LAST:event_jTextField10FocusLost

    private void jTextField10FocusGained(java.awt.event.FocusEvent
evt) { //GEN-FIRST:event_jTextField10FocusGained
        if ("Final".equals(jTextField10.getText()))
            jTextField10.setText("");
    } //GEN-LAST:event_jTextField10FocusGained

    private void jTextField9FocusLost(java.awt.event.FocusEvent evt)
{ //GEN-FIRST:event_jTextField9FocusLost
        if ("".equals(jTextField9.getText())) {
            jTextField9.setText("Inici");
        }
    } //GEN-LAST:event_jTextField9FocusLost

    private void jTextField9FocusGained(java.awt.event.FocusEvent evt)
{ //GEN-FIRST:event_jTextField9FocusGained
        if ("Inici".equals(jTextField9.getText()))
            jTextField9.setText("");
    } //GEN-LAST:event_jTextField9FocusGained

    private void jTextField8FocusLost(java.awt.event.FocusEvent evt)
{ //GEN-FIRST:event_jTextField8FocusLost
        if ("".equals(jTextField8.getText())) {
            jTextField8.setText("Exactitud");
        }
    } //GEN-LAST:event_jTextField8FocusLost

```

```

        private void jTextField8FocusGained(java.awt.event.FocusEvent evt)
        { //GEN-FIRST:event_jTextField8FocusGained
            if ("Exactitud".equals(jTextField8.getText()))
                jTextField8.setText("");
        } //GEN-LAST:event_jTextField8FocusGained

        private void jTextField7FocusLost(java.awt.event.FocusEvent evt)
        { //GEN-FIRST:event_jTextField7FocusLost
            if ("".equals(jTextField7.getText())) {
                jTextField7.setText("Residu (c0)");
            }
        } //GEN-LAST:event_jTextField7FocusLost

        private void jTextField7FocusGained(java.awt.event.FocusEvent evt)
        { //GEN-FIRST:event_jTextField7FocusGained
            if ("Residu (c0)".equals(jTextField7.getText()))
                jTextField7.setText("");
        } //GEN-LAST:event_jTextField7FocusGained

        private void jTextField3FocusLost(java.awt.event.FocusEvent evt)
        { //GEN-FIRST:event_jTextField3FocusLost
            if ("".equals(jTextField3.getText())) {
                jTextField3.setText("Abast (A)");
            }
        } //GEN-LAST:event_jTextField3FocusLost

        private void jTextField3FocusGained(java.awt.event.FocusEvent evt)
        { //GEN-FIRST:event_jTextField3FocusGained
            if ("Abast (A)".equals(jTextField3.getText()))
                jTextField3.setText("");
        } //GEN-LAST:event_jTextField3FocusGained

        private void jTextField11FocusGained(java.awt.event.FocusEvent
        evt) { //GEN-FIRST:event_jTextField11FocusGained
            if ("Sostre (c1)".equals(jTextField11.getText()))
                jTextField11.setText("");
        } //GEN-LAST:event_jTextField11FocusGained

        private void jTextField11FocusLost(java.awt.event.FocusEvent evt)
        { //GEN-FIRST:event_jTextField11FocusLost
            if ("".equals(jTextField11.getText())) {
                jTextField11.setText("Sostre (c1)");
            }
        } //GEN-LAST:event_jTextField11FocusLost

        public void setTiempo(int porcentaje){
            setError("Running "+porcentaje+"%");
            if (porcentaje==100){
                boton = false;
                archivo = false;
                setError("OK!");
            }
        }

        public void setTexto(String texto){
            setError(texto);
        }

        public static void setError(String error){
            jLabel3.setText(error);
        }

```

```

    public void prueba(XYDataset result) {
        archivo=false;
        JFreeChart chart =
ChartFactory.createScatterPlot("Semivariograma empíric", "Distància
(h)", "Covariància", result);
        ChartPanel cp = new ChartPanel (chart);
        cp.setMouseWheelEnabled(true);
        this.jPanel2.removeAll();
        this.jPanel2.add(cp, BorderLayout.CENTER);
        this.setVisible(true);
    }

    public void prueba2(XYDataset result) {
        ChartPanel cp = new ChartPanel (createChart(result));
        cp.setMouseWheelEnabled(true);
        this.jPanel2.removeAll();
        this.jPanel2.add(cp, BorderLayout.CENTER);
        //this.setVisible(true);
    }
    private static JFreeChart createChart(XYDataset result) {
        JFreeChart chart =
ChartFactory.createScatterPlot("Semivariograma empíric", "Distància
(h)", "Covariància", result);
        final XYPlot plot = chart.getXYPlot();
        plot.setBackgroundPaint(Color.lightGray);

        plot.setDomainGridlinePaint(Color.white);
        plot.setRangeGridlinePaint(Color.white);

        final XYLineAndShapeRenderer renderer = new
XYLineAndShapeRenderer();
        renderer.setSeriesLinesVisible(0, true);
        renderer.setSeriesLinesVisible(1, true);
        renderer.setSeriesLinesVisible(2, true);
        renderer.setSeriesShapesVisible(0, false);
        renderer.setSeriesShapesVisible(1, false);
        renderer.setSeriesShapesVisible(2, false);
        renderer.setSeriesLinesVisible(3, false);

        plot.setRenderer(renderer);
        return chart;
    }

    public void prueba3(XYZDataset result, double max, double min, int
multiplicar) {
        boton = false;
        if(max<0.1){
            max = (float) 0.1;
        }
        ChartPanel cp2 = new ChartPanel(createChart2(result,max,min,
multiplicar));
        cp2.setMouseWheelEnabled(true);
        this.jPanel5.removeAll();
        this.jPanel5.add(cp2, BorderLayout.CENTER);
        this.setVisible(true);
        if(jLabel3.getText().equals("Running 100%")){
            setError("OK!");
        }
    }
}

```



```

private static JFreeChart createChart2(XYZDataset dataset, double
maximo, double minimo, int multiplicar) {
    /* Nombre ejes */
    NumberAxis xAxis = new NumberAxis("X");
    xAxis.setAutoRangeIncludesZero(false);
    xAxis.setLowerMargin(0.0);
    xAxis.setUpperMargin(0.0);
    NumberAxis yAxis = new NumberAxis("Y");
    yAxis.setAutoRangeIncludesZero(false);
    yAxis.setLowerMargin(0.0);
    yAxis.setUpperMargin(0.0);

    yAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
    LookupPaintScale paintScale;
    XYBlockRenderer renderer = new XYBlockRenderer();
    try{
        paintScale = new
LookupPaintScale(minimo,maximo,Color.RED);
        Paint [] contourColors = getFullRainBowScale();
        double [] scaleValues = new double[contourColors.length];
        double delta = (maximo - minimo)/(contourColors.length -
1);

        double value = minimo;
        for(int i=0; i<contourColors.length; i++)
        {
            paintScale.add(value, contourColors[i]);
            scaleValues[i] = value;
            value = value + delta;
        }
    }catch(Exception e){
        paintScale = new LookupPaintScale();
        setError("Error: a massa petita");
    }
    renderer.setPaintScale(paintScale);

    XYPlot plot = new XYPlot(dataset, xAxis, yAxis, renderer);
    plot.setBackgroundPaint(Color.lightGray);
    plot.setDomainGridlinePaint(Color.white);
    plot.setRangeGridlinePaint(Color.white);
    plot.setForegroundAlpha(0.66f);
    plot.setAxisOffset(new RectangleInsets(5, 5, 5, 5));
    JFreeChart chart = new JFreeChart("Kriging", plot);
    chart.removeLegend();
    chart.setBackgroundPaint(Color.white);
    NumberAxis scaleAxis = new NumberAxis("Escala x"+multiplicar);
    scaleAxis.setTickLabelFont(new Font("Dialog", Font.PLAIN,
10));

    PaintScaleLegend psl = new
PaintScaleLegend(paintScale,scaleAxis);
    psl.setAxisOffset(5.0);
    psl.setPosition(RectangleEdge.BOTTOM);
    psl.setMargin(new RectangleInsets(5, 5, 5, 5));
    chart.addSubtitle(psl);
    return chart;
}

private static Paint[] getFullRainBowScale()
{
    int ncolor = 360;
    Color [] rainbow = new Color[ncolor];
    float x = (float) (1./(ncolor + 60.));
    for (int i=0; i < rainbow.length; i++)

```

```

        {
            rainbow[i] = new Color( Color.HSBtoRGB((i)*x,1.0F,1.0F));
        }
        return rainbow;
    }
    public Parametros getOpcion1(){
        param = new Parametros( opcion,

Float.parseFloat(jTextField3.getText()),

Float.parseFloat(jTextField7.getText()),

Float.parseFloat(jTextField11.getText()),
                                0,
                                0,
                                0);

        return param;
    }
    public Parametros getOpcion2(){
        param = new Parametros( opcion,

Float.parseFloat(jTextField3.getText()),

Float.parseFloat(jTextField7.getText()),

Float.parseFloat(jTextField11.getText()),

Integer.parseInt(jTextField8.getText()),

Integer.parseInt(jTextField9.getText()),

Integer.parseInt(jTextField10.getText()));
        return param;
    }
    public boolean getBoton(){
        return boton;
    }
    public boolean getArchivo(){
        return archivo;
    }
    public boolean getCampos1(){
        return (isNumber(jTextField3.getText()) &&
                (Float.parseFloat(jTextField3.getText())>0) &&
                (isNumber(jTextField7.getText()) &&
                (isNumber(jTextField11.getText()));
    }
    public boolean getCampos2(){
        return (isNumber(jTextField3.getText()) &&
                (Float.parseFloat(jTextField3.getText())>0) &&
                (isNumber(jTextField7.getText()) &&
                (isNumber(jTextField11.getText()) &&
                (isNumeric(jTextField8.getText()) &&
                (Integer.parseInt(jTextField8.getText())>0) &&
                (isNumeric(jTextField9.getText()) &&
                (isNumeric(jTextField10.getText()) &&

(Integer.parseInt(jTextField9.getText())<Integer.parseInt(jTextField10
.getText())));
    }
    public boolean isNumeric(String cadena){
        try {

```

```

        Integer.parseInt(cadena);
        return true;
    } catch (NumberFormatException nfe){
        return false;
    }
}
public LinkedList<Punto> getInput(){
    return puntos;
}
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay
with the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.ht
ml
    */
    /*
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(KrigingFrame.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(KrigingFrame.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(KrigingFrame.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(KrigingFrame.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    */
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new KrigingFrame().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;

```

```
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private static javax.swing.JLabel jLabel3;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel10;  
private javax.swing.JPanel jPanel11;  
private javax.swing.JPanel jPanel13;  
private javax.swing.JPanel jPanel14;  
private javax.swing.JPanel jPanel18;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JPanel jPanel4;  
private javax.swing.JPanel jPanel5;  
private javax.swing.JPanel jPanel8;  
private javax.swing.JPanel jPanel9;  
private javax.swing.JTextField jTextField10;  
private javax.swing.JTextField jTextField11;  
private javax.swing.JTextField jTextField3;  
private javax.swing.JTextField jTextField7;  
private javax.swing.JTextField jTextField8;  
private javax.swing.JTextField jTextField9;  
// End of variables declaration//GEN-END:variables  
private volatile boolean boton = false;  
private volatile boolean archivo = false;  
private volatile int opcion = 0;  
private Parametros param;  
private LinkedList<Punto> puntos = new LinkedList<>();  
}
```

ANNEX 2: DADES UTILITZADES EN ELS EXEMPLES

Dades del nivell de cadmi

Número	Coordenada X	Coordenada Y	Valor
1	288.0	311	11.5
2	285.6	288	08.5
3	273.6	269	07.0
4	280.8	249	10.7
5	273.6	231	11.2
6	276.0	206	11.6
7	285.6	182	07.2
8	288.0	164	05.7
9	292.8	137	05.2
10	278.4	119	07.2
11	360.0	315	03.9
12	355.2	291	09.5
13	367.2	272	08.9
14	367.2	250	11.5
15	352.8	226	10.7
16	350.4	203	08.3
17	369.6	180	06.1
18	369.6	165	06.7
19	357.6	139	06.2
20	355.2	118	00.0
21	434.4	312	05.5
22	451.2	295	04.0
23	448.8	268	07.0
24	432.0	252	05.3
25	441.6	228	11.6
26	441.6	204	09.0
27	444.0	182	14.5
28	441.6	160	12.1
29	432.0	140	00.9
30	444.0	119	00.0
31	254.4	172	03.2
32	254.4	128	01.2
33	254.4	299	01.7
34	333.6	301	01.2
35	333.6	271	07.6
36	333.6	194	11.6
37	333.6	163	08.7
38	412.8	285	05.8
39	254.4	257	03.8
40	412.8	172	10.4

41	412.8	150	10.0
42	492.0	282	07.1
43	492.0	249	04.4
44	492.0	315	10.4
45	492.0	150	01.6
46	444.0	190	15.0
47	436.8	240	03.4
48	360.0	195	06.8
49	345.6	210	10.8
50	254.4	216	14.9
51	280.8	216	09.9
52	307.2	216	11.6
53	333.6	216	06.5
54	360.0	216	10.1
55	386.4	216	11.8
56	412.8	216	11.0
57	439.2	216	16.7
58	465.6	216	11.6
59	492.0	216	06.9
60	345.6	216	09.9

Dades de l'aqüífer de Texas

Número	Coordenada X	Coordenada Y	Valor
1	4.278.275	12.762.282	1464
2	-2.739.691	9.078.732	2553
3	-116.289	8.489.600	2158
4	-1.861.823	7.645.199	2455
5	9.646.549	6.458.058	1756
6	10.856.243	8.292.325	1702
7	8.836.356	5.645.348	1805
8	9.004.213	3.925.820	1797
9	9.317.269	3.305.852	1714
10	9.761.099	5.627.887	1466
11	9.062.946	3.508.169	1729
12	9.255.262	4.175.238	1638
13	9.948.996	5.915.785	1736
14	-2.406.744	18.476.636	1476
15	-2.606.285	11.407.479	2200
16	5.627.842	2.684.826	1999
17	7.303.881	1.888.140	1680
18	8.026.679	1.261.593	1806
19	8.023.009	1.461.795	1682
20	6.883.845	10.777.423	1306

21	7.639.921	9.599.380	1722
22	6.446.148	11.039.641	1437
23	4.339.657	5.361.499	1828
24	3.907.769	6.199.805	2118
25	11.280.450	4.554.766	1725
26	5.425.899	14.781.987	1606
27	613.202	4.832.772	2648
28	-380.469	4.040.450	2560
29	-223.054	2.991.113	2544
30	-236.177	3.382.002	2386
31	-218.890	3.368.207	2400
32	6.322.428	7.949.924	1757
33	-1.077.860	17.511.346	1402
34	-1.898.889	17.191.694	1364
35	-3.857.884	15.852.742	1735
36	8.314.496	15.911.558	1376
37	-2.180.248	1.502.551	2729
38	-2.356.457	941.441	2766
39	-2.011.299	2.209.269	2736
40	-1.662.654	1.725.621	2432
41	2.990.748	17.512.875	1024
42	10.091.568	2.297.808	1611
43	10.129.544	2.296.385	1548
44	10.326.625	2.034.239	1591
45	-1.431.073	3.126.545	2540
46	-1.813.447	3.018.118	2352
47	-1.812.151	2.953.241	2528
48	-988.796	3.814.483	2575
49	-1.216.336	3.911.081	2468
50	1.165.754	1.873.347	2646
51	6.169.122	3.249.406	1739
52	6.957.896	3.380.841	1674
53	6.672.205	3.393.264	1868
54	-3.665.446	15.091.456	1865
55	-1.955.102	13.778.404	1777
56	-2.129.791	13.182.542	1579
57	-2.236.166	13.713.680	1771
58	2.114.719	13.926.199	1408
59	768.461	12.683.751	1527
60	-83.327	10.777.691	2003
61	5.670.724	17.126.443	1386
62	5.900.052	16.454.863	1089
63	6.896.893	17.724.820	1384
64	7.090.225	16.138.136	1030
65	7.300.243	16.298.960	1092

66	5.966.237	17.010.544	1161
67	6.187.429	17.430.178	1415
68	6.370.810	17.391.453	1231
69	562.706	7.908.730	2300
70	1.824.739	7.739.191	2238
71	8.568.824	13.981.701	1038
72	10.507.646	13.203.181	1332
73	-10.164.278	1.065.106	3510
74	-14.523.654	2.802.333	3490
75	-7.399.313	8.797.270	2594
76	-9.448.182	8.662.606	2650
77	-8.884.983	7.670.991	2533
78	-12.025.898	8.076.485	3571
79	-8.602.454	5.436.334	2811
80	-7.279.097	4.309.215	2728
81	-10.017.372	4.289.881	3136
82	-7.883.539	4.082.141	2553
83	-8.369.063	4.650.482	2798
84	-9.561.661	3.582.183	2691
85	-8.755.480	2.939.267	2946